INTERNET PROGRAMMING AND WEB DESIGN

DIRECTORATE OF DISTANCE EDUCATION

MBA

Paper 4.4.2



ALAGAPPA UNIVERSITY

Karaikudi - 630 003 Tamil Nadu

Reviewer	
Prof. M. Selvam	Dean, Faculty of Management Alagappa University, Karaikudi

Authors

Kalapatapu Kalyani: Units (2, 3) © Kalapatapu Kalyani, 2013

VIkas Publishing House: Units (1, 4, 5, 6) © Reserved, 2013



Vikas® is the registered trademark of Vikas® Publishing House Pvt. Ltd.

VIKAS® PUBLISHING HOUSE PVT. LTD.
E-28, Sector-8, Noida - 201301 (UP)
Phone: 0120-4078900 • Fax: 0120-4078999
Regd. Office: 576, Masjid Road, Jangpura, New Delhi 110 014
• Website: www.vikaspublishing.com • Email: helpline@vikaspublishing.com

Work Order No.: AU/DDE/D2/Printing/31/2013-14 Date: 14.03.2013 Copies 1000

SYLLABI-BOOK MAPPING TABLE

Internet Programming and Web Design

Syllabi	Mapping in Book
UNIT 1 **Foundations for Internet Programming:** An Overview of Internet Programming, Web System Architecture, URL, Domain Name System, Overview of HTTP, HTTP Request-Response, Generation of Dynamic Web Pages Cookies, WWW Design Issues, Security and Encryption, Developing Intranet Applications.	**Unit 1:** Foundations for Internet Programming **(Pages 3-32)**
UNIT 2 **Internet Programming Languages:** Java, Java in Windows, Common Gateway Interface (CGI), Perl: Strawberry Perl or Active Perl, Microsoft Internet Implementation.	**Unit 2:** Internet Programming Languages **(Pages 33-62)**
UNIT 3 **Internet Scripting Languages:** JavaScript, VB Script, Other Scripting Languages, Java Basics, I/O Streaming, Files, Looking Up Internet Address, Socket Programming, Client-Server Programs, E-Mail Client, SMTP, POP3 Programs, Web Page Retrieval, Protocol Handlers, Content Handlers, Applets, Image Handling, Remote Method Invocation.	**Unit 3:** Internet Scripting Languages **(Pages 63-104)**
UNIT 4 **Internet Markup Languages:** HTML & SGML: Introduction, Basic HTML, Formatting and Fonts, Commenting Code, Anchors, Backgrounds, Images, Hyperlinks, Lists, Tables, Frames, Simple HTML Forms, XHTML, Netscape Extensions, Microsoft Internet Explorer, Only HTML Tags, Shockwave and Lingo, Dynamic HTML: Introduction, Cascading Style Sheets.	**Unit 4:** Internet Markup Languages **(Pages 105-172)**
UNIT 5 **ActiveX Controls:** Creating an ActiveX Control to Activate a Web Page, VDOLive Technology, Creating Netscape Navigator Plug-Ins, Pulling Web Information, Creating a Custom Integrated Application with Multiple Protocols.	**Unit 5:** ActiveX Controls **(Pages 173-242)**
UNIT 6 **Web Graphics:** A Graphical View of Web, Essay Web Graphics, Images and Hyperlinks, Adding Graphics to Web Pages, Site and Page Design, Framing Your Graphics, Dynamic Graphics, Animation.	**Unit 6:** Web Graphics **(Pages 243-278)**

CONTENTS

INTRODUCTION

NOTES

Web technology revolutionizes the way information is exposed and circulated within every organization and enables everyone to be in the distribution business. It includes the development, hosting and maintenance of sites for big, medium and small companies. Nowadays every organization is building a Website for the World Wide Web (WWW) on the Internet. Most organizations are on a network be it LAN, WAN or MAN. With Web technology and a network, every organization can build their own Websites that are available to everyone on the authorized network. To build a Website one requires a computer system, a network, browser software and server software. The network must support the TCP/IP protocol and each computer must have an IP address configured on it. The Internet provides free Web browser, Web server software, icons, backgrounds, images and graphics to develop an attractive Website. A Web server contains a bunch of files that are called by the server software when a user sends a request through their Web browser. The Web browsers provide accessibility to various available Websites to have the latest information online.

Programming languages enable you to create custom applications and add functionality to it. On the Internet, programming languages enable you to create visual animation, respond to user actions, validate forms and interact with databases to provide solutions. Programming languages are of two types, namely interpreted and compiled. The scripting languages are mostly interpreted, which means that you write the code and the browser or server understands what to do with it. Compiled languages are generally used for server side processing for Web applications. Programming technologies generally use one or more programming language to create end-to-end solutions for Web applications.

The Internet is a global network that uses the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide. A Web browser is a type of software that retrieves and presents information resources on the Internet. In order for the Internet to be a viable form of communication and information, standard are required for the related technologies and languages. Three Internet standard organizations exist to define technologies, specifications and languages, and to develop standards for global communication networks, including the Internet and networking protocols. The World Wide Web Consortium (W3C) provides standards, specifications and guidelines for technologies that are commonly used on the Internet. The Internet, as its name suggests, is an interconnection of networks at an international level. The World Wide Web (WWW) is an architectural framework for accessing linked documents spread out over thousands of machines all over the Internet. WWW made it possible for a site to set up a number of pages of information containing text, pictures, sound and even video with embedded links to other pages. It is a graphic variety of online hypertext encyclopedia that lets you browse through the Internet's vast resources simply by clicking with your mouse.

Web technologies are referred as the interface between the Web servers and their clients. There are several Web technologies, from simple to complex, which include markup languages, programming interfaces and languages, and standards for document identification and display.

NOTES

This book, *Internet Programming and Web Design*, is aimed at giving the students a fair thought of the foundations of the Internet programming and markup languages, the basics of Web design and the basics of the Internet programming using various languages to create a Web page with the help of graphics and animation. The book comprises six units dealing with the foundations for the Internet programming, WWW design issues, developing intranet application, the Internet programming languages (Java, CGI, Perl), scripting languages (JavaScript, VBScript), SMTP, uses of applets, introduction to HTML and SGML, creating an ActiveX control to activate a page and the basics of Web design.

The book follows the self-instruction format wherein each unit begins with an 'Introduction' to the topic of the unit followed by an outline of the 'Unit Objectives'. The detailed content is then presented in a simple and structured form interspersed with 'Check Your Progress' questions to facilitate a better understanding of the topics discussed. The 'Key Terms' are given on respective pages to help the student revise what he/she has learnt. A 'Summary' along with a set of 'Questions and Exercises' is also provided at the end of each unit for effective recapitulation.

UNIT 1 FOUNDATIONS FOR INTERNET PROGRAMMING

NOTES

Structure

1.0 INTRODUCTION

In this unit, you will learn about the basics of the Internet. The Internet is a 'network of networks'. It can be visualized as a number of interconnected computers located at physically disparate locations around the world. It is a global network of computers, cutting across barriers and boundaries of countries, class or race. It comprises of thousands of computer networks linked to each other. The Internet carries a vast array of information resources and services, most notably the interlinked hypertext documents of the World Wide Web (WWW) and the infrastructure to support electronic mail. In this unit, you will also learn about the various features of the Internet like Telnet, file transfer protocol, uniform resource locator, etc. Finally, you will learn about intranet which is a standard term used for a set of private computer networks operating within an organization. The intranet network technologies are used to ease communication between people or work groups of an organization and also to enhance the data sharing capacity of an organizational database.

1.1 UNIT OBJECTIVES

After going through this unit, you will be able to:

- Understand the basics of the Internet programming
- Explain how the Internet works
- State the various Internet protocols
- Define the significance of URL and URL encoding
- Describe the significance of HTTP in the functioning of WWW
- Explain the significance of World Wide Web
- Define why intranet applications are developed

NOTES

1.2 AN OVERVIEW OF INTERNET PROGRAMMING

Traditionally, computers were standalone systems. Transferring information from one computer to another could be done through mediums, such as the floppy disk and tape. Resources, such as printers, scanners and Compact Disk Read Only Memory or CD-ROM drives could not be shared across the computers. They had to be attached physically to the computers to use them. However, with the invention of networking, it became possible to share resources across computers. In addition, it enabled sharing the knowledge base thereby helping people to take advantage of a rich information base and work collectively.

The Internet is a 'network of networks'. It can be visualized as a number of interconnected computers located at physically disparate locations around the world. It is a global network of computers, cutting across barriers and boundaries of countries, class, race or sex. It comprises of thousands of computer networks linked to each other.

These computer networks are categorized into the following types:

Local Area Network or LAN

Local Area Network or LAN connect devices, such as computers and printers over short distances typically within an office premises or a campus. Computers and devices are connected through cables within a distance of 2000 feet. Wireless Local Area Network (WLAN) and Bluetooth technology are also types of LANs.

Wide Area Network or WAN

Wide Area Network or WAN span over areas larger than a single building or a campus. They can span over multiple buildings or different offices of an organization across physically different locations in the same city or across cities.

Metropolitan Area Network or MAN

Metropolitan Area Network or MAN refers to the LANs connected through high speed, seamless inter connection within a 'metropolitan' area. The word 'metropolitan' does not necessarily mean a city but can be any area that is spread out but is treated as one entity for example, a company having its buildings across a large area of land. MAN covers geographical areas that are larger than LAN but smaller than WAN.

1.2.1 Major Features of the Internet

The Internet has made communication simple. It can be used for the following:

- **Direct Communication:** You can send messages to family and friends, business associates and acquaintances using the electronic mail facility. Using electronic mail or e-mail, you can send and receive messages within a few seconds anywhere in the world. Using Internet Relay Chat (IRC) you can have online communication with people over the Internet. You can log into a chat room and converse with others by typing messages that are instantly delivered. With the improvement of network technologies and increase in broadband, not only you can send text messages but you can also send graphics, audio and video to other people.

NOTES

- **Online Shopping:** Logically, the Internet has removed all barriers of distance and nationality. You can shop for products and services across the world by logging on to a Web portal. You can also pay your bills online using credit and debit cards. You can also transfer money between different accounts with the click of a mouse.
- **Distance Education:** The Internet provides a perfect medium for knowledge sharing and information dissemination. Courses are available on the Internet. You can register and pay online, and can complete a course on different interest areas. You can also pursue specialized higher studies in the comfort of your home.
- **Knowledge Base:** The Internet provides a rich information base that can be accessed by people around the globe. In fact, it is one of the richest information bases that can be accessed with the click of a mouse. Using search engines, you can search for detailed information on any topic of your interest.
- **Banking:** Banks are using information technology to provide online banking facilities to their customers. Using the Internet, you can now view your account details, get drafts made, request for cheque books and transfer money from one account to another. The use of Automated Teller Machines or ATMs has shifted the mundane back office work to the customer himself. Instead of hiring bank clerks in abundant, banks are using ATMs to considerably reduce time and operational costs.
- **Travel:** Using the Internet, travel agencies can provide their services on the Web along with the latest discounts, packages and availability details so that customers can compare rates, make online bookings and avail the discounts without having to run around multiple offices.
- **Bill Payment:** The government sector has also realized the benefits of Information Technology or IT. Now, you can make online payments for public utilities, such as water, electricity and phones, using credit cards as the payment medium.

1.2.2 Working of the Internet

You can use the following options for connecting to the Internet.

Direct Connection

Through a direct connection, a machine is directly connected to the Internet backbone and acts like a gateway. Though a direct connection provides full access to all the Internet services, it is very expensive to implement and maintain. Direct connections are suitable only for very large organizations or companies.

Through Internet Service Provider or ISP

You can also connect to the Internet through the gateways provided by the Internet Service Providers or ISPs. The range of the Internet services varies depending on the ISPs. Therefore, you should use the Internet services of the ISP that is best suitable for your requirements. You can connect to your ISP using two methods which are as follows:

NOTES

Remote Dial-Up Connection

A dial-up connection (see Figure 1.1) enables you to connect to your ISP using a modem. A modem converts the computer bits or digital signals to modulated or analog signals that the phone lines can transmit and vice versa. Dial-up connection uses either SLIP (Serial Line Internet Protocol) or PPP (Point-to-Point Protocol) for transferring information over the Internet.

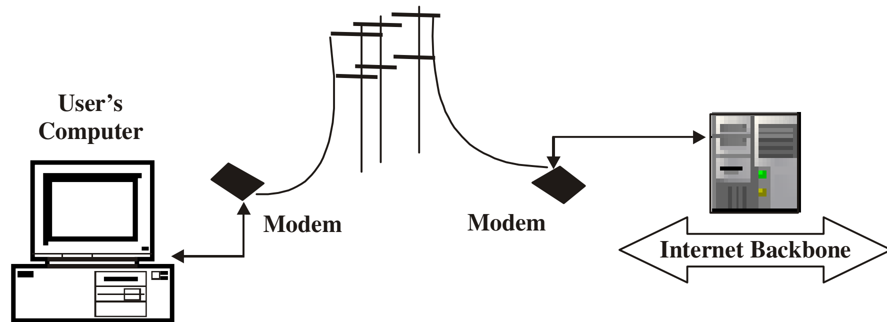


Fig. 1.1 *A Dial-Up Connection*

For dial-up connections, regular telephone lines are used. Therefore, the quality of connection is not always good.

Permanent Dedicated Connection



Dedicated Internet connection: A permanent telephone connection between two points

You can also have a dedicated Internet connection that typically connects you to ISP through a dedicated phone line. A **dedicated Internet connection** is a permanent telephone connection between two points. Computer networks that are physically separated are often connected using leased or dedicated lines. These lines are preferred because these are always open for communication traffic unlike the regular telephone lines that require a dialing sequence to be activated. Often this line is an ISDN (Integrated Services Digital Network) line that allows transmission of data, voice, video and graphics at very high speeds. ISDN applications have revolutionized the way business communicates. ISDN lines support upward scalability which means that you can transparently add more lines to get faster speeds – going up to 1.28 Mbps (Million bits per second).

T1 and T3 are the other two types of commonly used dedicated line types for the Internet connections T1 line is made up of fiber optic, which can carry nearly 60 times more data than any normal residential modem can carry. The T3 lines are made up of 28 T1 lines, which normally operate at a signaling rate of 1.544 Mbit/s. Dedicated lines are becoming popular because of their faster data transfer rates. Dedicated lines are cost effective for the business that uses the Internet services extensively.

The Internet Protocols

The Internet protocols are required to transfer data over the networks and/or the Internet in an efficient manner. When various computers are connected through a computer network, it becomes necessary to use a protocol to efficiently use network bandwidth and avoid collisions.

A network protocol defines a language that contains rules and conventions for reliable communication between different devices over the network. For example, it includes rules that specify how to package data into messages, how to acknowledge a message and how to compress data.

A number of the Internet protocols are used nowadays. The most commonly used protocols are as follows:

- Transmission Control Protocol/Internet Protocol (TCP/IP)
- HyperText Transfer Protocol (HTTP)
- File Transfer Protocol (FTP)
- Telnet

NOTES

Transmission Control Protocol/Internet Protocol or TCP/IP

It is a protocol suite used to transfer data over the Internet. Two main protocols in this protocol suite are as follows:

TCP: It forms the higher layer of TCP/IP and divides a message or a file into smaller packets which are transmitted over the Internet. A TCP layer on the other side receives these packets and reassembles the data packets into the original message.



TCP: It forms the higher layer of TCP/IP and divides a message or a file into smaller packets which are transmitted over the Internet

IP: It is the lower layer whose function is to handle the address part of each packet to enable it to be delivered to the right destination. Usually, this address is checked by each gateway computer on the network so that it is identified where to forward the message. This implies that all the packets of a message are delivered to the destination regardless of the route used for delivering the packets.

The working of TCP/IP can be compared with shifting your residence to a new location. This activity would involve packing your belongings in smaller boxes for easy transportation with the new address and a number written on each of the boxes. You would then load them on multiple vehicles. These vehicles may take different routes to reach the same destination. The delivery time of vehicles depends on the amount of traffic and the length of the route. Once the boxes are delivered to the destination, you would check these to make sure that all have been delivered in good shape. After that, you need to unpack the boxes and reassemble your house.

HyperText Transfer Protocol or HTTP

HTTP is a protocol for transferring files (text, image, sound, video and other multimedia files) using the Internet. It is an application protocol that runs on top of the TCP/IP protocol suite which is the foundation protocol of the Internet. It defines how messages are formatted and transmitted and what actions Web servers and Web browsers should take in response to the commands issued. HTTP is based on a client-server architecture where your Web browser acts as a HTTP client making requests to the Web server machines.



HTTP: It is a protocol for transferring files (text, image, sound, video and other multimedia files) using the Internet.

File Transfer Protocol or FTP

FTP is an application protocol used for exchanging files between computers over the Internet. It is the simplest and most commonly used protocol for downloading/ uploading a file from/to a server. For example, downloading a document or an article from a Web site. Like other protocols, FTP also uses TCP/IP protocol suite for data transfer.

FTP also works on a client-server architecture where an FTP client program is used to make a request to an FTP server (files can be stored on computers referred to as FTP servers). Basic FTP support is usually provided as a part of the TCP/IP suite of programs.

NOTES



IP address: An IP address is a unique number associated with each computer making it uniquely identifiable amongst all the computers connected to the Internet

Telnet

Telnet is a protocol that allows you to access a remote computer provided you have been given the permission to do so. It is typically referred to as 'remote login'. Telnet is based on a different concept from HTTP and FTP. HTTP and FTP allows you to request specific files from remote computers only while with Telnet you log in as a regular user on a remote machine with whatever privileges you may have been granted on that computer to access specific application and data. A request for a connection to a remote host which may be a computer lying physically in a neighbouring room or miles away results in an invitation to log in with a user id and a password. If the request is accepted, the user can enter commands through the Telnet program and these would be executed as if they were being entered directly from the host machine. Once connected, the user's computer emulates the remote computer. Telnet is typically used by program developers or anyone who needs to access data and/or applications located at a particular host computer.

The Internet Addressing

Since the Internet consists of a large number of computers connected with each other, it requires a proper addressing system to uniquely identify each computer in the network. Each computer connected to the Internet is associated with a unique number and/or a name called computer address. Before you wish to access any Web page on a computer, you would require the computer address.

IP Address: An IP address is a unique number associated with each computer making it uniquely identifiable amongst all the computers connected to the Internet. This is a 32-bit number and is divided into four octets, such as 00001010 00000000 00000000 00000110. For human readability, it is represented in a decimal notation, separating each octet with a period. The above number would therefore be represented as: 10.0.0.6.

Each octet can range from 0 – 255, thus all IP addresses lie between 0.0.0.0 to 255.255.255.255 resulting in total 4294, 967,296 possible IP addresses. It may be worth noting that no two machines (or hosts) can have the same IP address.

Domain Naming System or DNS: In a network, computers and devices can be grouped as a unit with common rules and procedures. Such a group is called a domain. Each domain name therefore corresponds to a group of IP addresses. Some examples of domain names include Yahoo.com, Microsoft.com and abcuniv.edu. The last portion of the domain name provides information on the kind of organization to which the address belongs.

Commonly used abbreviations in domain names are summarized in Table 1.1.

Table 1.1 *Abbreviations used in Domain Names*

Abbreviation	*What it Represents*
com	Commercial organization
org	Non-profit organization
edu	Educational institution
net	Networking organizations
gov	Government agencies

DNS is an Internet service that translates domain names to or from IP addresses, which is the actual basis of addressing on the Internet. A DNS is typically a database containing information about domain names and their corresponding IP addresses (see Figure 1.2).

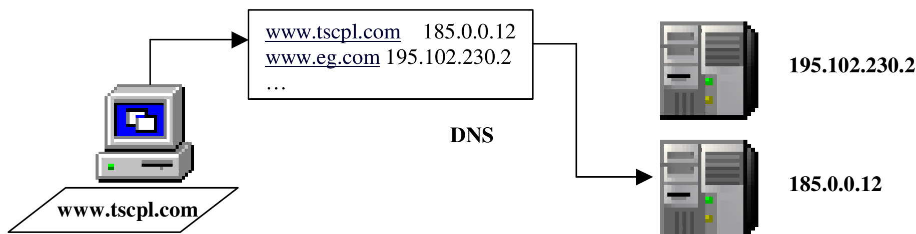


Fig. 1.2 *DNS*

NOTES

HyperText Markup Language or HTML

HTML is the language for publishing hypertext or HTML pages on WWW. It is a non-proprietary format. You can use a large number of tools for creating HTML pages, ranging from simple editors to sophisticated authoring tools that work on What You See Is What You Get or WYSIWYG principle. HTML provides various tags to structure text in an HTML page. For example, you can use `<p></p>`, `<li></li>` and `<a></a>` tags to structure text into paragraphs, lists and hypertext links.



HTML: It is the language for publishing hypertext or HTML pages on WWW

1.3 URL

URL denotes Uniform Resource Locator. It is the address of a document on the World Wide Web. Web browsers enable a person to enter either a known address in the Web server or a specific document within that server. Addresses usually begin with http://, ftp://, gopher://, wais://, file://, etc. It is not feasible to maintain WWW without using the URLs. These are also used to represent hypermedia links and links to network services within the HTML documents. Any file or service on the Internet can be presented with the help of the URL. The first part of the URL that comes before the two slashes specifies the method of access or protocol being followed for communications between the browser and the Web server. The second part coming after two slashes represents the address of the host machine, whose data or services are being sought. The remaining parts signify the names of the files, the port to connect to or the text to search for in a database. All the parts of an address for obtaining a file or service from a host machine in a URL are shown as a single unbroken line with no spaces and the locations of the host machines or Websites that run www servers are typically named with a www at the beginning of the network address. The Web browsers enable the users to access Web services by specifying a URL and connecting to that document or service. Once the user gets connected with the Web server, the Web browsers select the hypertext in an HTML document and send a request to open a URL. Thus, hyperlinks are used not only to provide other texts and media in the same document but also to facilitate other network services. Web browsers are not simply Web clients. They are full-featured FTP, Gopher and telnet clients.



URL: It is the address of a document on the World Wide Web

NOTES

URL Encoding

Uniform Resource Locator (URL) is a pointer that avails specified resources across the net. A Resource simply connotes information containing files or directories. It is referenced with query to available databases through search engines, such as Google or Yahoo. An example of URL that appears on the address bar is given as follows:

http://aaa.bbb.edu/flower.html

Table 1.2 depicts the above given URL details:

Table 1.2 *URL Details*

URL part	**Function**
http	The protocol specifier
aaa.bbb.edu	The domain name
flower.html	The page location

The http is used as a protocol in which information resides on the domain called **aaa.bbb.edu.** The information that resides in the host machine is taken as flower.html. The host machine can either be protocol dependent or host dependent. A component of URL is known as the path component. Sometimes the URL is also referred to as ‘Port’, that is, it is a port number by which TCP connection is possible to the remote host machine. The default port for protocol is used if port is not specified. For instance, Port 80 is known as default port for HTTP. The two ports, Port 20 and Port 21 are used by ftp. The alternative port can be used in the following way:

http://aaa.bbb.edu:80/ flower.html

Table 1.3 shows some specific symbols and characters which are used by the URL. These are, in fact, URL encoding.

Table 1.3 *Used Symbols and URL Encoding*

Specific Symbols and Characters	**URL Encoding**
;	%3B
?	%3F
/	%2F
:	%3A
#	%23
&	%24
=	%3D
+	%2B
$	%26
,	%2C
%	%25
<	%3C
>	%3E
~	%7E
%	%25
<space>	+ or %20

Note: The <spacebar> is frequently used and ‘+’ sign is reserved for its URL encoding. For example, string ‘A B’ in URL is encoded as either ‘A%20B’ or ‘A+B’.

Check Your Progress

1. Define the term Internet.
2. What does MAN stands for?
3. State any two features of the Internet.
4. What is URL?
5. What is path component?

1.4 OVERVIEW OF HTTP

NOTES

The engine of the World Wide Web is the application protocol that defines how Web servers and clients exchange information using the *HyperText Transfer Protocol (HTTP)*. The first version of HTTP, HTTP/0.9, was part of the early World Wide Web and was a very simple request/response protocol with limited capabilities that could transfer only text files. The first widely used version was HTTP/1.0, which is a more complete protocol that allows the transport of many types of files and resources. The current version is HTTP/1.1, which expands HTTP/1.0's capabilities with several features that improve the efficiency of transfers and address many of the needs of the rapidly growing modern World Wide Web.

The HyperText Transfer Protocol is a protocol specifically designed to allow the transfer of HyperText Markup Language (HTML) documents. HTML is a tag language used to create HyperText documents. HyperText documents include links to other documents that contain additional information about the highlighted term or subject. Such documents can contain other elements apart from text, such as graphic images, audio and video clips, Java applets and even virtual reality worlds.

Overview of HTTP

HTTP is based on request-response activity. A client, running an application called a browser, establishes a connection with a server and sends a request to the server in the form of a request method. The server responds with a status line, including the message's protocol version and a success or error code, followed by a message containing server information, entity information and possible body content.

An HTTP transaction is divided into the following four steps:

1. The browser opens a connection.
2. The browser sends a request to the server.
3. The server sends a response to the browser.
4. The connection is closed.

On the Internet, HTTP communication generally takes place over TCP connections. The default port is TCP 80, but other ports can be used. This does not preclude HTTP from being implemented on top of any other protocol on the Internet or on other networks. HTTP only presumes a reliable transport; any protocol that provides such guarantees can be used.

HTTP is a stateless protocol because it does not keep track of the connections. To load a page including two graphics, for example, a graphic-enabled browser will open three TCP connections: one for the page and two for the graphics. Most browsers, however, are able to handle several of these connections simultaneously.

HTTP Operation

In most cases, the HTTP communication is initiated by the user agent requesting a resource on the origin server. In the simplest case, the connection is established through a single connection between the user agent and the origin server (see Figure 1.3).

NOTES

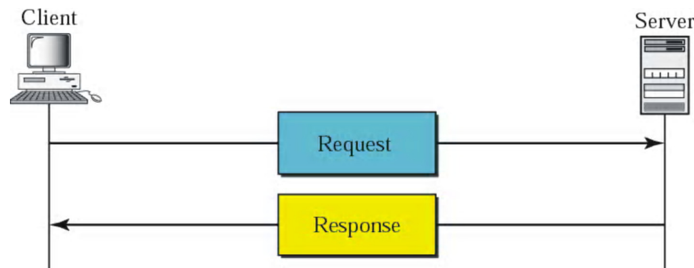


***Fig. 1.3** HTTP Operation*

HTTP is a client-server oriented and request-response protocol. Basic communication consists of an *HTTP Request* message sent by an *HTTP Client* to an *HTTP server*, which returns an *HTTP Response* message back to the client.

The simple client-server operational model of HTTP is complicated when *intermediary devices*, such as proxies, tunnels or gateways are inserted in the communication path between the HTTP client and server. HTTP/1.1 is specifically designed with features to support the efficient conveyance of requests and responses through a series of steps from the client through the intermediaries to the server and back again. The entire set of devices involved in such a communication is called the *Request-Response Chain* (see Figure 1.4).

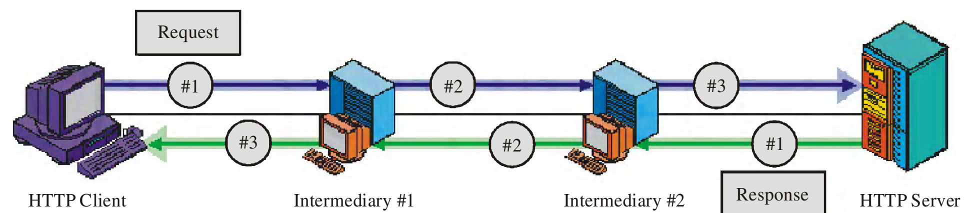


***Fig. 1.4** HTTP Operation with Proxies*

HTTP Message

All the communication between devices using the HyperText Transfer Protocol takes place via *HTTP messages*. There are only two types of messages: *Requests* and *Responses*. Clients usually send requests and receive responses, while servers receive requests and send responses. Intermediate devices, such as gateways or proxies may send and receive both types of message.

HTTP General Message

All HTTP messages are text-based messages created to fit a message structure that the standard calls the *Generic Message Format*.

HTTP messages consist of the following fields:

Message Types

A HTTP message can be either a client request or a server response. The following string indicates the HTTP message type:

HTTP Message = Request/Response

Message Header

The HTTP message header field can be one of the following:

- General header
- Request header

NOTES

- Response header
- Entity header
- Message body

Message body can be referred to as entity body if there is no transfer coding. Message body simply carries the entity body of the relevant request or response.

Message length indicates the length of the message body if it is included.

HTTP Request Message Format

HTTP requests are the means by which HTTP clients ask servers to take a particular type of action, such as sending a file or processing user input (see Figure 1.5).

The structure of HTTP requests:

```
<request-line>
<general-headers>
<request-headers>
<entity-headers>
<empty-line>
[<message-body>]
[<message-trailers>]
```



Fig. 1.5 *HTTP Request*

Each request message begins with a *request line*, containing three critical pieces of information: the *method* (type of action) the client is requesting; the URI (Uniform Resource Identifier) of the resource upon which the client wishes the action to be performed, and the version of HTTP that the client is using. After the request line come a set of message headers related to the request, followed by a blank line and then optionally, the message body of the request.

- **General Headers:** General headers refer mainly to the message itself, as opposed to its contents and are used to control its processing or provide the recipient with extra information. They are not particular to either request or response messages, so they can appear in either. They are likewise not specifically relevant to any entity the message may be carrying.
- **Request Headers:** These headers convey to the server more details about the nature of the client's request and give the client more control over how the request is handled. For example, special request headers can be used by the client to specify a conditional request—one that is only filled if certain criteria are met. Others can tell the server which formats or encodings the client is able to process in a response message.

- **Entity keaders:** These are headers that describe the entity contained in the body of the request, if any.

HTTP Response Message Format

NOTES

Each request message sent by an HTTP client to a server prompts the server to send back a *response message* (see Figure 1.6).

The structure of HTTP response:

```
<status-line>
<general-headers>
<response-headers>
<entity-headers>
<empty-line>
[<message-body>]
[<message-trailers>]
```

HTTP/1.1 200 OK	Status Line	HTTP Response
Date: Thu, 20 May 2004 21:12:58 GMT Connection: close	General Headers	
Server: Apache/1.3.27 Accept-Ranges: bytes	Response Headers	
Content-Type: text/html Content-Length: 170 Last-Modified: Tue, 18 May 2004 10:14:49 GMT	Entity Headers	
<html> <head> <title>Welcome to the Amazing Site!</title> </head> <body> This site is under construction. Please come back later. Sorry! </body> </html>	Message Body	

Fig. 1.6 *HTTP Response*

Each response message starts with a *status line* that contains the server's HTTP version number and a numeric *status code* and text *reason phrase* that indicate the result of processing the client's request. The message then contains headers related to the response, followed by a blank line and then the optional message body. Since most HTTP requests ask for a server to return a file or other resource, many HTTP responses carry an entity in the message body (see Figure 1.7).



General headers: General headers refer to the message itself and are not specific to response messages or the entity in the message body

- **General Headers:** General headers refer to the message itself and are not specific to response messages or the entity in the message body. These are the same as the generic headers that can appear in request messages (though certain headers appear more often in responses and others are more common in requests).
- **Response Headers:** These headers provide additional data that expands upon the summary result information in the status line. The server may also return extra result information in the body of the message, especially when an error occurs, as discussed below.
- **Entity Headers:** These are headers that describe the entity contained in the body of the response, if any. These are the same entity headers that can appear in a request message, but they are seen more often in response messages. The

reason for this is simply that responses more often carry an entity than requests, because most requests are to retrieve a resource.

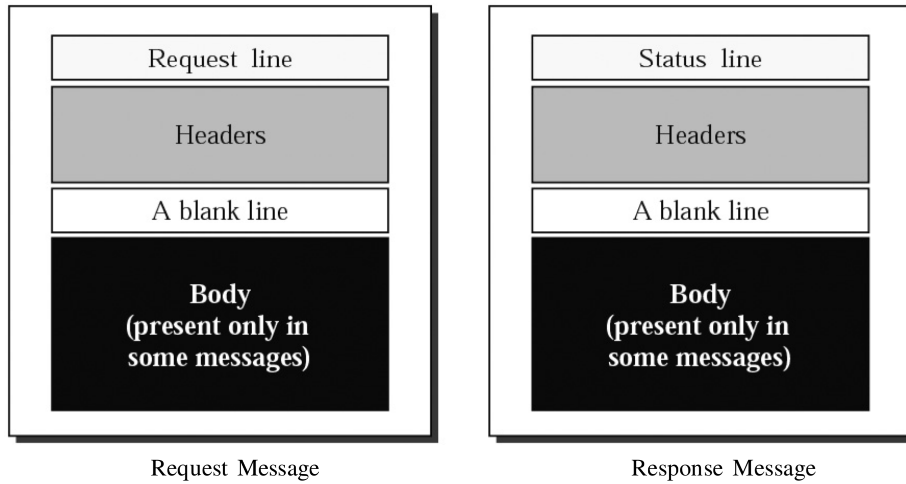


Fig. 1.7 *HTTP Messages*

NOTES

Request and Status Lines

Figure 1.8 (a) and (b) illustrates the structure of request line and status line, respectively.

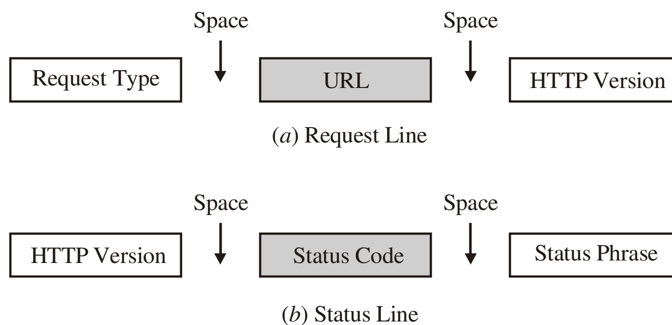


Fig. 1.8 *Request and Status Lines*

Status Codes

Table 1.4 defines the various status codes that are generally displayed to the users.

Table 1.4 *Status Codes*

Code	*Phrase*	*Description*
	Informational	
100	Continue	The initial part of the request has been received and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
	Success	
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon.
204	No content	There is no content in the body.

NOTES

Redirection		
301	Multiple choices	The requested URL refers to more than one resource.
302	Moved permanently	The requested URL is no longer used by the server.
304	Moved temporarily	The requested URL has moved temporarily.
Client Error		
400	Bad request	There is a syntax error in the request.
401	Unauthorized	The request lacks proper authorization.
403	Forbidden	Service is denied.
404	Not found	The document is not found.
405	Method not allowed	The method is not supported in this URL.
406	Not acceptable	The format requested is not acceptable.
Server Error		
500	Internal server error	There is an error, such as a crash, at the server site.
501	Not implemented	The action requested cannot be performed.
503	Service unavailable	The service is temporarily unavailable, but may be requested in the future.

Method Definitions

Currently defined methods are as follows:

Safe and Idempotent Methods

Methods considered not to cause side effects are referred to as safe. Idempotent methods are GET, HEAD, PUT and DELETE.

- **OPTIONS:** This method allows the client to determine the options or requirements associated with a source or capabilities of a server without any resource retrieval.
- **GET:** This method allows the client to retrieve the data that was determined by the request URI.
- **HEAD:** This method allows the client to retrieve meta information about the entity that does not require you to transfer the entity body.
- **POST:** The post function is determined by the server.
- **PUT:** This method is similar to the post method with one important difference: The URI in post request identifies the resource that will handle enclosed entity.
- **DELETE:** This methods requests that the server delete the source determined by the request URI.
- **TRACE:** This method allows the client to see how the message was retrieved at the other side for testing and diagnostic purposes.

HTTP Message Headers

Much of the functionality in HTTP is actually implemented in the form of *message headers*, which convey important details between clients and servers. Some headers can appear in only HTTP requests, some in only HTTP responses and some in either type of message (see Figure 1.9).

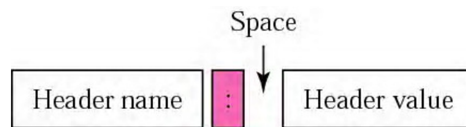


Fig. 1.9 HTTP Message Header

NOTES

HTTP General Headers

HTTP *general headers* are so named because unlike the other three categories, they are not specific to any particular kind of message or message component (request, response or message entity) (see Table 1.5). General headers are used primarily to communicate information about the message itself, as opposed to what content it carries. They provide general information and control how a message is processed and handled.

Table 1.5 *Types of Headers*

Header	*Description*
Cache-control	Specifies information about caching
Connection	Shows whether the connection should be closed or not
Date	Shows the current date
MIME-version	Shows the MIME version used
Upgrade	Specifies the preferred communication protocol

HTTP Request Headers

HTTP *request headers* are used only in HTTP request messages and serve a number of functions in them (see Table 1.6).

- They allow the client to provide information about itself to the server.
- They give additional details about the nature of the request that the client is making.
- They allow the client to have greater control over how its request is processed and how (or even if) a response is returned by the server or intermediary.



HTTP request headers: They are used only in HTTP request messages and serve a number of functions in them

Table 1.6 *Request Headers*

Header	*Description*
Accept	Shows the media format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
From	Shows the e-mail address of the user
Host	Shows the host and port number of the client
If-modified-since	Send the document if newer than specified date
If-match	Send the document only if it matches given tag
If-non-match	Send the document only if it does not match given tag
If-range	Send only the portion of the document that is missing
If-unmodified-since	Send the document if not changed since specified date
Referrer	Specifies the URL of the linked document
User-agent	Identifies the client program

NOTES

HTTP Response Headers

The counterparts to request headers, response headers appear only in HTTP responses (see Table 1.7). They provide additional data that expands upon the summary information that is present in the status line at the beginning of each server reply. Many of the response headers are sent only in response to the receipt of specific types of requests, or even particular headers within certain requests.

Table 1.7 *Response Headers*

Header	*Description*
Accept-range	Shows if server accepts the range requested by client
Age	Shows the age of the document
Public	Shows the supported list of methods
Retry-after	Specifies the date after which the server is available
Server	Shows the server name and version number

HTTP Entity Headers

These headers provide information about the resource carried in the body of an HTTP message, called an *entity* in the HTTP standards (see Table 1.8). They serve the overall purpose of conveying to the recipient of a message the information it needs to properly process and display the entity, such as its type and encoding method. At least one entity header should appear in any HTTP message that carries an entity. However, they may also be present in certain responses that do not have an actual entity in them. Most notably, a response to a *HEAD* request will contain all the entity headers associated with the resource specified in the request; these are the same headers that would have been included with the entity, had the *GET* method been used instead of *HEAD* on the same resource. Entity headers may also be present in certain error responses, to provide information to help the client make a successful follow-up request.

Table 1.8 *Entity Headers*

Header	*Description*
Allow	Lists valid methods that can be used with a URL
Content-encoding	Specifies the encoding scheme
Content-language	Specifies the language
Content-length	Shows the length of the document
Content-range	Specifies the range of the document
Content-type	Specifies the media type
Etag	Gives an entity tag
Expires	Gives the date and time when contents may change
Last-modified	Gives the date and time of the last change
Location	Specifies the location of the created or moved document

Some Other Features

- The most important feature that improves the efficiency of operation of HTTP is *caching*—the storing of recently requested resources in a temporary area. If the same resource is then needed again a short time later, it can be retrieved from the cache rather than requiring a fresh request to the server, resulting in a savings of both time and bandwidth. Caching can be performed by Web clients, servers and intermediaries. The closer the cache is to the user, the greater the efficiency benefits; the farther from the user, the greater the number of users that can benefit from the cache.

NOTES

- One of the most important types of intermediary devices in HTTP is a *proxy server*, which acts as a middleman between the client and server, handling both requests and responses. A proxy server may either transport messages unchanged or may modify them to implement certain features and capabilities. Proxies are often used to increase the security and/or performance of Web access.

Generation of Dynamic Web Page Cookies

HTTP is an inherently *stateless* protocol, because a server treats each request from a client independently, forgetting about all prior requests. This characteristic of HTTP is not an issue for most routine uses of the World Wide Web, but is a problem for interactive applications such as online shopping where the server needs to keep track of a user's information over time. To support these applications, most HTTP implementations include an optional feature called *state management*. When enabled, a server sends to a client a small amount of information called a *cookie*, which is stored on the client machine. The data in the cookie is returned to the server with each subsequent request, allowing the server to update it and send it back to the client again. Cookies thus enable a server to remember user data between requests. However, they are controversial, because of certain potential privacy and security concerns related to their use.

1.5 WWW

The World Wide Web or WWW is a global hypertext system that was initially developed in 1989 by Tim Berners Lee at the European Laboratory for Particle Physics, CERN, in Switzerland to facilitate an easy way of sharing and editing research documents among a geographically dispersed group of scientists.

The WWW has a unique combination of flexibility, portability and user friendly features that distinguish it from other features provided by the Internet.

Architecture of WWW

Figure 1.10 illustrates the architecture of WWW which supports the interoperability design principle, assured by the implementation of compatible languages and protocols, and enables access, exchange and processing of information among agents. The evolution of the WWW technology supports maintainability and decentralization of information systems.

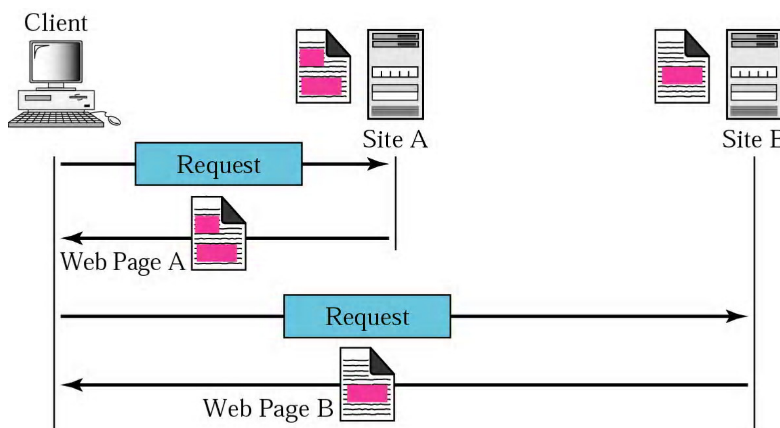


Fig. 1.10 *Architecture of the World Wide Web*

Check Your Progress

6. Define the term HTTP protocol.
7. Classify the types of messages in HTTP.
8. What is a general header?
9. Name any three request headers used in HTTP.
10. Which is the most important feature that improves the efficiency of operation of HTTP?

NOTES

Functional Components of the World Wide Web

The World Wide Web is a complete system that is comprised of a number of related components, of which HTML (HyperText Markup Language), HTTP (HyperText Transfer Protocol) and URI (Uniform Resource Identifier) are most essential. HTML describes how hypertext documents are constructed. HTML allows links between documents to be represented while the HTTP is the application layer protocol that moves hypertext and other documents over the Web. The URI is a mechanism which provides a consistent means of identifying resources, both on the Web and more generally on the Internet.

Figure 1.11 illustrates the major functional components (HTML, HTTP, URIs, etc.) of WWW.

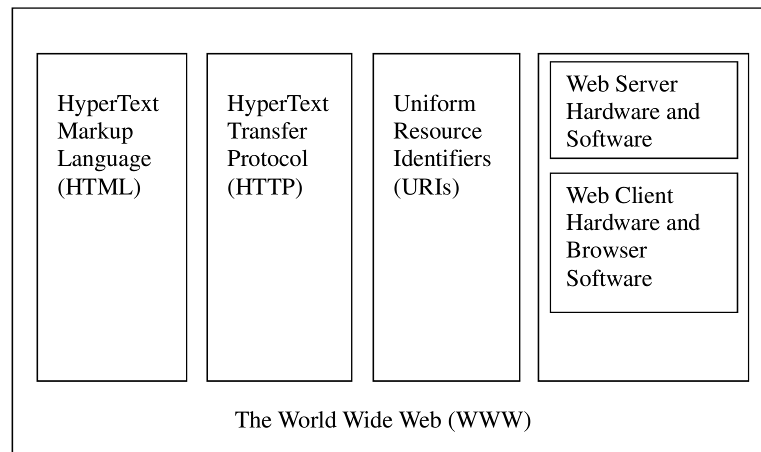


***Fig. 1.11** Major Components of WWW*

Web Browsers



Web browsers: HTTP client software programs that run on TCP/IP client computers to access Web documents on Web servers

Web browsers are HTTP client software programs that run on TCP/IP client computers to access Web documents on Web servers. These browser programs retrieve hypertext documents and display them, and also implement many of the Web's advanced features, such as caching (see Figure 1.12). Browsers used today support a wide variety of media, allowing the Web to implement many different functions aside from simply hypertext document transfer. Examples include displaying images, playing sounds and implementing interactive programs.

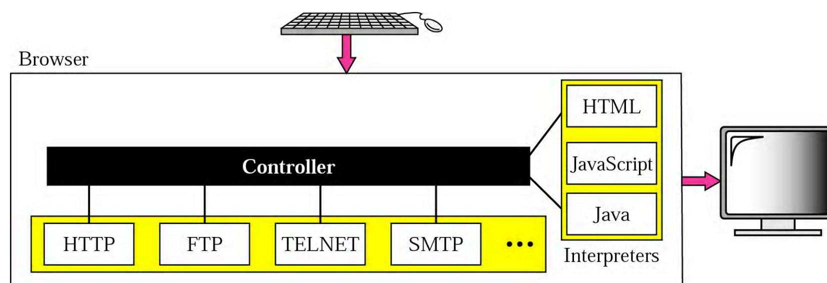


***Fig. 1.12** Architecture of a Web Browser*

Each browser usually consists of three parts, i.e., a controller, client protocol, and interpreters. The controller receives input from the keyboard or the mouse and uses client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on screen.

NOTES

Web Servers

Web servers are computers that run special server software to allow them to provide hypertext documents and other files to clients who request them. Millions of such machines around the world now serve as a virtual, distributed repository of the enormous wealth of information that the Web represents.

Working of WWW

Following are the working phenomena of WWW:

- Viewing a Web page on the WWW normally begins either by typing the URL of the page into a Web browser or by following a hyperlink to that page or resource.
- First, the server name portion of the URL is resolved into an IP address using the global, distributed Internet database known as the domain name system.
- The browser then requests the resource by sending an HTTP request to the Web server at that particular address.
- The HTML text of the page is requested first and parsed immediately by the Web browser which will then make additional requests for images and any other files that form a part of the page.
- Having received the required files from the Web server, the browser then renders the page onto the screen as specified by its HTML, Cascading Style Sheets or CSS and other Web languages. Any images and other resources are incorporated to produce the on-screen Web page that the user sees.

Hypertext

Hypertext is the main concept that makes the WWW more than just another message transfer system. The prefix 'hyper' usually means 'above' or 'beyond' and thus hypertext is like text, but goes beyond it in terms of functionality. The extra information in a hypertext document is used to tell the computer program that displays the file to a user how to format it. This information takes the form of special instructions that are interspersed with the actual text of the document itself which are written according to the syntax of a defining language. This addition of extra elements to the content of a document is commonly called *marking up* the document.

The WWW hypertext documents use *HyperText Markup Language (HTML)*. HTML documents are as American Standard Code for Information Interchange or ASCII text files, but are arranged using a special structure of HTML *elements* that define the different parts of the document and how they should be displayed to the user. Each element is described using special text *tags* that define it and its characteristics (see Figure 1.13).

< TagName	Attribute = Value	Attribute = Value	•••	>

a. Beginning tag

< /TagName >

b. Ending tag

Fig. 1.13 *HTML Tag*

NOTES



Static documents: These are fixed content documents that are created and stored in a server

Web Documents

The documents in WWW can be grouped into three broad categories.

Static Documents

These are fixed content documents that are created and stored in a server. The client can only get a copy of the document. The contents of the file are determined when it is created and not when it is used. The user cannot change the document (see Figure 1.14).

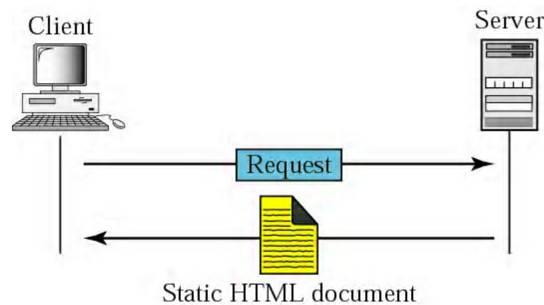


***Fig. 1.14** Static Document*

Dynamic Documents

This document is created by the server whenever the browser requests the document. When a request arrives, the Web server runs an application program or a script that creates the dynamic document (see Figure 1.15). The server returns the output of the program or script as response to browser that requested the document (see Figure 1.16). A fresh document is created for each request; the contents of dynamic document can vary from one request to another for example, retrieval of date and time from a server.

There are two ways to create dynamic documents.

- Common Gateway Interface (CGI) is a technology that creates and handles dynamic documents. CGI is a set of standards that defines how a dynamic document is created, how data is input to the program and how output result is used.
- Scripting technologies are embedded in the HTML page.

  Example: PHP, JSP, ASP, etc.

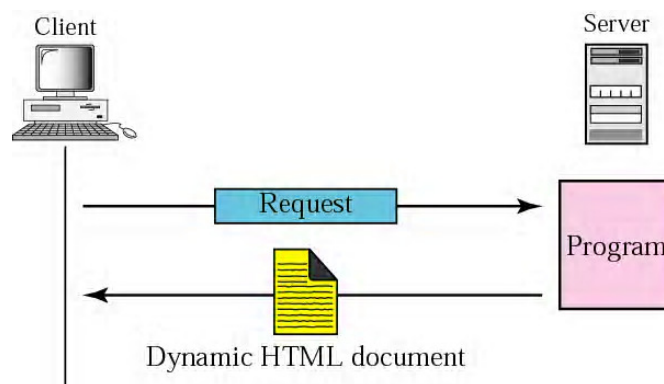


***Fig. 1.15** Dynamic Document with Application Program*

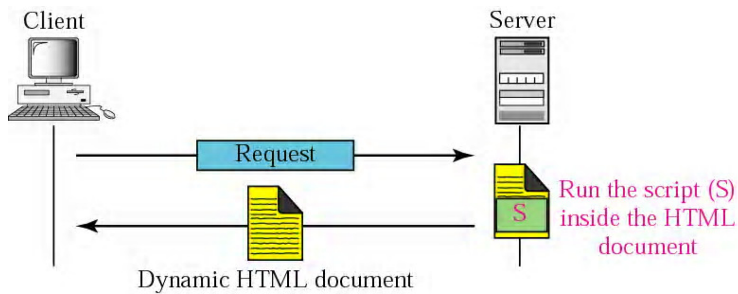


Fig. 1.16 *Dynamic Document with Script*

NOTES

Active Documents

In active documents the program or script runs at the client side. When a browser requests an active document, the server sends a copy of the document or script. The document is then run on the client side. Active documents are sometimes referred to as client-side dynamic documents (see Figure 1.17).



Active documents: In active documents the program or script runs at the client side

Active documents can be created in two ways:

- Java applets, programs written in Java on the server are compiled and ready to run. The browser creates an instance of this applet and runs it.

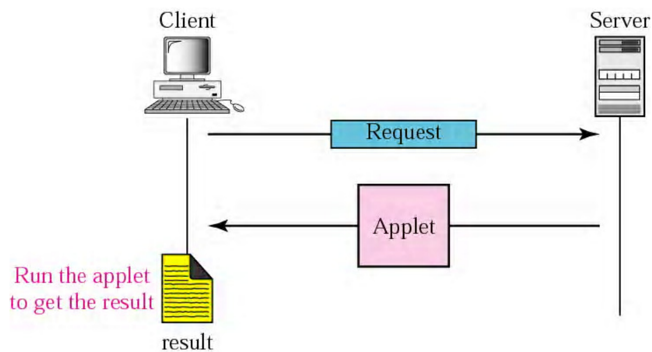


Fig. 1.17 *Active Document*

- JavaScript is interpreted and run by the client at the same time. The script is in the source code (see Figure 1.18).

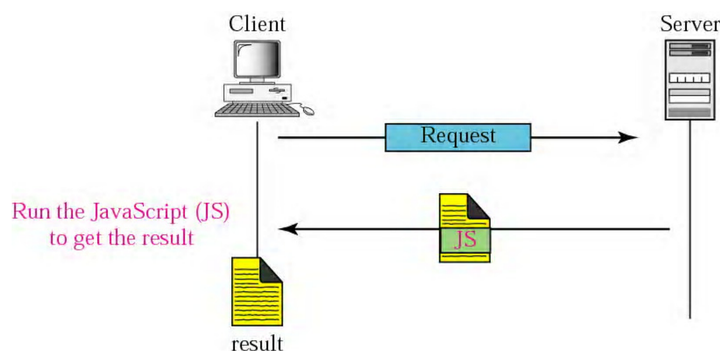


Fig. 1.18 *Active Document*

WWW Design Issues, Security and Encryption

The World Wide Web Consortium (W3C) is the main international standards organization for the World Wide Web (abbreviated WWW or W3). Founded by Tim

NOTES

Berners-Lee at MIT and currently headed by him, the consortium is made up of member organizations for the development of standards for the World Wide Web.

Great numbers of users are tapping into the WWW or simply the Web for using graphic interfaces and relatively inexpensive access to unlimited information. The Web can be an excellent infrastructure for any dynamic application as it uses Graphical User Interface (GUI). Infrastructure refers to the facts that the Internet provides through the Web browser interface to specified application(s). The architectural principle specifications must be according to the guidelines specified by W3C. The following criteria and features must be considered for design specifications:

- Intended uses of the system.
- Availability on which platforms?
- Navigational techniques and tools: browsing, indexing, maps, resource discovery, etc.
- Multiuser access that must support protection, editing, locking and annotation.
- The topology of the Web of links.

The following are three important issues which require agreement between systems which work together:

- Naming and addressing of documents.
- Protocols.
- The format in which node content is stored and transferred.
- Implementation and optimization - Caching smart browsers, format conversion, gateways, etc.

There are several issues related to the public Internet. In the Internet environment, the application must be designed securely so that the network or private TCP/IP network (intranet) functions appropriately. Hence, a high-quality system development methodology is required for the Internet. Starting with requirements definition, the detailed technical design, construction and testing the Internet portion of specific application must be designed methodologically.

First determine which components will be Internet based because all components of the application can not be Internet based. Hence, in the application design determine which modules of the application will be Internet based. Because the timing and security are key issues of using the Internet, hence modules requiring strict security or where timing is critical should not be implemented on the Internet.

By designing an interface enables the users to take advantage of specified tools. The users of the system must know how to access the Internet by using their Web browsers, Web mail and File Transfer Protocol (FTP). Introducing new screens, new addresses and new filenames is an incremental change in known procedures for potential users. This setup is especially important while designing an online application. HyperText Markup Language (HTML) is derived from Standard Generalized Markup Language (SGML, formally called ISO 88791) used in the publishing industry. HTML, together with URLs (Universal Resource Locators) and HTTP, is one of the foundations of the World Wide Web.

NOTES

Some Web browsers implement information packets called *cookies* as a mechanism to store information about a host directory tree. Cookies can be used to store key information that can later be used to retrieve information stored on the host.

Timing Issues: Timing can be a problem on the Internet. The user does not know how much transaction time it will take, several seconds or even much longer. Hence, timing-critical portions of the application should not be Internet based. The design of the application affects the way in which timeouts are determined. Commercial Web browsers have built-in timeout settings.

Security

Security is a major issue with the Internet because it is public domain. The public nature of the Internet can cause security concerns that do not exist for private intranet or dial-up applications. Because packets pass through machines over which there is no control, someone can potentially see confidential information. Any hacker with a network data scope can get credit card numbers, Social Security numbers and other confidential information from your transmissions. Hence, you must specifically design for these potential security leaks.

Encryption

Many types of encryption can be used to protect the transactions. Several Web browsers and hosts are 'secure' because they encrypt information passing between them. The extent to which the encryption can be used in the application will depend on the sensitivity of the information and the cost of encryption. For example, if you are writing your own application in which you will provide both the client and server modules then you have to provide your own custom encryption schemes.

While designing an application to be hosted by a Web server, always place the application on a secure Web server. These servers establish a secure connection with the client browser and encrypt all information that passes between them. For example, the Secure Sockets Layer (SSL) is used by Netscape Server to encrypt pages during transmission.

1.6 DEVELOPING INTRANET APPLICATIONS

Intranet is the standard term used for a set of private computer networks operating within an organization. The intranet network technologies are used to ease communication between people or work groups of an organization and also to enhance the data sharing capacity of an organization database. The intranets use standard network hardware and software technologies, such as Ethernet, WiFi, TCP/IP, Web browsers and Web servers. Technically, an organization's intranet supports Internet access but it is firewalled in such a secured way that the computers connected to its network cannot be accessed directly from the outside. In several organizations, intranets are shielded and safeguarded from unauthorized external access through a network gateway and firewall. Basically, extranets which are the general extension to intranets releases this firewall for providing organized and secured access to outsiders. Extranets broaden a

NOTES

private network onto the Internet with distinctive specifications for Authentication, Authorization and Accounting (AAA) protocol. Thus, an intranet is principally a corporate productivity tool. Organization may permit access to off-site employees, if required, through a virtual private network or via additional access methods by defining unique user authentication and encryption.

A simple intranet computer network system includes an internal e-mail system and possibly a message board service. More advanced and sophisticated intranet networks include Websites and companies databases in the form of news, forms and employee's personnel information. In addition, for e-mail and groupware applications, an intranet usually integrates company's internal Websites, documents and/or databases.

Technically, within an organization an intranet computer network uses Internet Protocol (IP) technology for sharing information within functional systems or computers and may include multiple Local Area Networks (LANs). One of the standard and popular Internet protocols can be defined in an intranet, such as HTTP (HyperText Transfer Protocol) for Web services, SMTP (Simple Mail Transfer Protocol) for e-mail facilities and FTP (File Transfer Protocol) for easy exchange of information from secured resources. Specific Internet technologies can be set for providing current or advanced interfaces to legacy information systems that host corporate data of an organization. Such an intranet is considered as a private analog of the Internet or as a private extension of the Internet restricted to an organization.

Nowadays, intranets are uniquely set to provide relevant tools and applications to enhance productivity in an organization, such as collaboration that assists while working in groups and during teleconferencing, organizing corporate directories using advanced sales and customer relationship management tools, and analysing project management. Intranets are also widely used as platform to revolutionize corporate culture, such as employees can converse over the key issues of their organization via an intranet forum application which may give innovative ideas to management for productivity, quality and other corporate issues.

Planning and Creation of Intranet

The intranet in an organization has strategic importance in the success and growth of an organization. The following are some significant points for planning an intranet:

- Planning the objective and goals of the intranet.
- Specifying persons or departments accountable for the implementation and management of the intranet.
- Specifying functional strategies, information architecture, page layouts and design of the intranet.
- Determining implementation schedules and phase out of existing systems, if any.
- Defining and implementing security of the intranet.
- Ensuring it within legal boundaries and other constraints.
- Defining desired level of interactivity, for example wikis, online forms, etc.

NOTES

- Checking the input of new data and updating of existing data so that it can be centrally controlled or devolved.

Usually, intranets are considered as static sites. Fundamentally, they are shared drives specifically set to operate on centrally stored documents. However, organizations are maintaining their intranets in such a way that it has become a 'communication hub' for their groups/employees and specifically support the feature 'socializing' intranets. The authentic implementation includes the following steps:

Step 1: Securing senior management support and funding.

Step 2: Business requirements analysis.

Step 3: Identifying users' information needs.

Step 4: Installation of Web server and user access network.

Step 5: Installation of required user applications on computers.

Step 6: Creation of document framework for the content to be hosted.

Step 7: User involvement in testing and promoting use of intranet.

Step 8: Ongoing measurement and evaluation, including through benchmarking against other intranets.

Step 9: Selecting key personnel for maintaining the intranet and keeping content up-to-date.

Intranet Development

While planning a secured intranet, the most important point is to assign the responsibility. Depending on the size of organization, the responsibility can be assigned to single employee or a group of employees who will further determine what content should be published on the intranet and what technology will be used in the process. This intranet team will consider the size, cost and functionality of the intranet. The following are some universal considerations which must be considered when planning for an organization intranet:

- Who is the primary audience of intranet? All employees? Or only certain departments?
- What are the goals of intranet? Improved document access? Better collaboration? Cut down on printing costs?
- How will the success of the intranet be measured?
- What types of documents and which corporate databases will be accessed?
- What Web-based applications will be accessible from the intranet?
- How will the site be structured? What will be on the homepage, the different main landing pages and sub-pages?
- Will everyone be allowed to publish content to the intranet or only certain employees?
- Who will be in-charge of communicating editorial guidelines and maintaining editorial consistency?

Check Your Progress

11. List the major components of WWW.
12. Define the term Web browser.
13. In how many categories can the Web documents be classified?
14. What are static documents?
15. State any three components required for setting up a secure and reliable intranet.

Besides, the intranet team must address the basic technical questions for smooth functioning of the intranet during the planning phase. Some of the significant ones are as follows:

- Who will be in-charge of configuring and maintaining the Web server?
- Will the server and network be administered by in-house information technology staff or by outside contractors?
- What security precautions (firewalls, security software) are taken to limit the access to the intranet?
- How much network bandwidth will the intranet require?
- How will new applications be tested before being added to the intranet?
- What Content Management System (CMS) will be used to create and publish content?
- How will employees be trained on the CMS?
- Who will be accountable for technical issues/questions about the CMS?
- How will the company back up the intranet data? How does intranet data fit into the company's larger disaster recovery plan?

After settling all these issues, the intranet team presents a proposed budget to the executives' in-charge. In bigger organizations, these may be the Chief Technical Officer (CTO) and/or Chief Information Officer (CIO). In the intranet budget, the cost projections for the following items must be included:

- Web servers.
- People to administer servers (in-house or contractors).
- Web development and design (in-house or contractors).
- Content management system.
- Application development (software and personnel).
- Security hardware and software.
- Long term maintenance costs.

After the approval of the budget, the team can set up a safe and effective intranet.

Setting up an Intranet

For setting up a secure and reliable intranet, the following components are required:

- Web server (hardware and software).
- Networked personal computers.
- Firewall hardware and software.
- Content Management Software (CMS).
- Other application software.

In a Web server, the hardware helps in using the intranet Web server and depends on the size of the intranet, the content to be published and the number of people accessing the intranet at any given time. The server software handles all requests for files hosted on the server, finds the file and sends it off to the right computer.

NOTES

Check Your Progress

16. Fill in the blanks with appropriate words.
    (a) The most important feature that improves the efficiency of operation of HTTP is ___________.
    (b) The prefix '___________' usually means 'above' or 'beyond'.
    (c) In a____________ documents the program or script is running at the client side.
    (d) ______________ is the standard term used for a set of private computer networks operating within an organization.
17. State whether the following statements are true or false.
    (a) MAN covers geographical areas that are larger than LAN but smaller than WAN.
    (b) It is always feasible to maintain WWW without using the URLs.
    (c) HTTP request headers are used only in HTTP request messages and serve a number of functions in them.
    (d) HyperText is the main concept that makes the WWW more than just another message transfer system.

Thus, development of intranet applications is a layered process. First, the technical infrastructure should be created. Next the end solutions that attract users to the technology must be employed. The tools that make intranet technology manageable and provide more efficient development environments must be established.

NOTES

1.7 SUMMARY

- The Internet is a 'network of networks'. It can be visualized as a number of interconnected computers located at physically disparate locations around the world.
- Local Area Network or LAN connects devices, such as computers and printers over short distances typically within an office premises or a campus.
- Wide Area Network or WAN span over areas larger than a single building or a campus.
- Metropolitan Area Network or MAN refers to the LANs connected through high speed, seamless inter connection within a 'metropolitan' area.
- In direct communication you can send messages to family and friends, business associates and acquaintances using the electronic mail facility.
- Through a direct connection, a machine is directly connected to the Internet backbone and acts like a gateway.
- The range of the Internet services varies depending on the ISPs.
- Transmission Control Protocol/Internet Protocol or TCP/IP is a protocol suite used to transfer data over the Internet.
- IP is the lower layer whose function is to handle the address part of each packet to enable it to be delivered to the right destination.
- HTTP is a protocol for transferring files (text, image, sound, video and other multimedia files) using the Internet.
- FTP is an application protocol used for exchanging files between computers over the Internet.
- FTP also works on a client-server architecture where an FTP client program is used to make a request to an FTP server.
- Telnet is a protocol that allows you to access a remote computer provided you have been given the permission to do so.
- An IP address is a unique number associated with each computer making it uniquely identifiable amongst all the computers connected to the Internet.
- URL denotes Uniform Resource Locator. It is the address of a document on the World Wide Web.
- The HyperText Transfer Protocol is a protocol specifically designed to allow the transfer of HyperText Markup Language (HTML) documents. HTML is a tag language used to create HyperText documents.

- HTTP is a client-server oriented and request-response protocol. Basic communication consists of an HTTP Request message sent by an HTTP client to an HTTP server, which returns an HTTP Response message back to the client.
- HTTP requests are the means by which HTTP clients ask servers to take a particular type of action, such as sending a file or processing user input.
- General headers refer mainly to the message itself, as opposed to its contents, and are used to control its processing or provide the recipient with extra information.
- HTTP request headers are used only in HTTP request messages and serve a number of functions in them
- The WWW has a unique combination of flexibility, portability and user friendly features that distinguish it from other features provided by the Internet.
- Web browsers are HTTP client software programs that run on TCP/IP client computers to access Web documents on Web servers.
- Web servers are computers that run special server software to allow them to provide HyperText documents and other files to clients who request them.
- HyperText is the main concept that makes the WWW more than just another message transfer system.
- The Intranet in an organization has strategic importance in the success and growth of an organization.

1.8 ANSWERS TO 'CHECK YOUR PROGRESS'

1. The Internet is a 'network of networks'. It can be visualized as a number of interconnected computers located at physically disparate locations around the world.
2. MAN stands for Metropolitan Area Network.
3. The two features of Internet are direct communication and online shopping.
4. URL is the address of a document on the World Wide Web.
5. A component of URL is known as the path component.
6. The HyperText Transfer Protocol is a protocol specifically designed to allow the transfer of HyperText Markup Language (HTML) documents. HTML is a tag language used to create HyperText documents.
7. A HTTP message can be either a client request or a server response. The following string indicates the HTTP message type:

   HTTP Message = Request/Response.
8. General headers refer to the message itself and are not specific to response messages or the entity in the message body.
9. The three request headers that are used in HTTP are Aceept, From and Host.

10. The most important feature that improves the efficiency of operation of HTTP is caching.
11. The major components of WWW are HTML, HTTP, URIs, Web server hardware and software, and Web client hardware and browser software.
12. Web browsers are HTTP client software programs that run on TCP/IP client computers to access Web documents on Web servers.
13. The documents in WWW can be classified into three categories, namely dynamic documents, static documents and active documents.
14. Static documents are fixed content documents that are created and stored in a server.
15. For setting up a secure and reliable intranet, the following components are required:
    - Web server (hardware and software).
    - Networked personal computers.
    - Firewall hardware and software.
16. (a) Caching, (b) Hyper, (c) Active, (d) Intranet.

    (a) True, (b) False, (c) True, (d) True.

NOTES

1.9 QUESTIONS AND EXERCISES

Short-Answer Questions

1. Define the significance of the Internet.
2. List the major applications of the Internet.
3. Define the term HTTP.
4. What is URL encoding?
5. Write down the structure of HTTP Response message format.
6. Differentiate between HTTP general headers and HTTP request headers.
7. Define the significance of active documents in WWW.
8. What is an intranet?
9. What are the points that should be kept in mind while planning an intranet?
10. List the important components required for setting up an intranet.

Long-Answer Questions

1. What is an Internet? Describe the various features of Internet.
2. Explain the significance and utilities of HyperText protocol or HTTP with reference to WWW.
3. What is URL encoding? Discuss the various symbols and characters used by the URL.

NOTES

4. Describe the HTTP operation with the help of a diagram.
5. What are the request and statues lines? Explain with the help of a diagram.
6. Explain the term WWW and its architecture with the help of a diagram.
7. Write a note on Web documents.
8. Discuss the important points associated with the planning and creation of intranet.

UNIT 2 INTERNET PROGRAMMING LANGUAGES

NOTES

Structure

2.0 INTRODUCTION

In this unit, you will learn about the basics of the Internet programming languages, Java and Perl. Java was developed as a platform independent language for consumer electronic devices but later on it evolved as a preferred language for the Internet. Java is platform independent language which means that the applications created in Java can run on different operating systems. Java is a portable language that generates an intermediate code, which is converted into machine code by Java Virtual Machine (JVM). Various features of Java, such as simple, platform independent, portable, object oriented, secure and robust enables to develop robust, portable and manageable applications. A programming paradigm (also known as programming methodology) describes the structure of a program. In other words, it determines how the instructions are placed in a program. You will learn to create Java programs. Finally, you will learn about the Perl programming language. You will also learn about basic Perl variables and data types. A variable is a place to store a value, so you can refer to it or manipulate it throughout your program. Perl has three types of variables, namely scalars, arrays and hashes. You will also learn about CGI environment variables which is a series of hidden values that the Web server sends to every CGI program you run.

NOTES

2.1 UNIT OBJECTIVES

After going through this unit, you will be able to:

- Explain the history and features of Java
- Discuss different data types and variables of Java
- Explain Java operators and their precedence
- Discuss the various control statements used in Java
- Explain the basic structure of Perl
- Describe how to start Perl programs
- Comprehend Perl variables
- Discuss CGI environment variables

2.2 OVERVIEW OF JAVA

If you already know C, you only have to make a little more effort to become a Java expert. However, if you have learned C++, you have to unlearn a little before you proceed to learn Java.

Java, which is an object oriented programming language is based on the concept of an object. If you have some familiarity with an object oriented language, such as Smalltalk or C++, much of Java will be familiar to you. Java is derived from C++, but is slightly simplified with libraries and convenient for the Internet. It is the programming language for the Internet environment. The Internet implies heterogeneous systems, different network features, different windows libraries and different operating systems. Java guarantees identical program behaviour on different platforms.

Java is an interpreted language. Java compiler complies the Java source code to obtain an object code termed as bytecode in Java terminology. The Java interpreter will interpret and run by the byte code at runtime.

Java is architectural neutral language. Architectural neutrality means that the bytecode that is the output of the Java compiler will run on any processor and operating system. You can compile a Java program to obtain the bytecode. This bytecode or object code will run on any processor and operating system, provided a suitable interpreter is available.

Java is portable as well. This means that Java provides the same Application Programmer's Interface (API) functions or library calls on each system. Java API functions are the same for a programmer on Window 95, Windows NT or on Solaris 2.x. To port a program from one of these platforms to another, what the programmer generally does is to recompile the program. However, due to the architectural neutrality of Java, you do not even have to do that.

Why Java is Important to the Internet?

Java came to the forefront of programming with the help of the Internet. Consequently, Java had a deep effect on the Internet. In addition to simplifying Web programming in

general, Java innovated a new type of networked program called the Applet. A Java Applet is a small program embedded in a Web page which runs when that page is browsed using a Web browser.

NOTES

Java's Magic: The Bytecode

You have just learned that the output of a Java compiler is not an executable code. Rather, it is a bytecode. In Java bytecode is a highly optimized set of instructions planned to be executed by the Java runtime system, which is referred to as the Java Virtual Machine (JVM). A Java program executed by the JVM helps solve problems associated with Web based programs. Since the JVM is in control, it can hold the program and check it from generating side effects outside the system.

Java Buzzwords

As mentioned earlier, portability and security is the innovation of Java. Some additional factors too played significant roles in moulding the final form of Java. The list of buzzwords summed up by the Java team is as follows:

- **Simplicity:** Programmers find it easy to write applications as it avoids the use of the concepts of C/C++, such as pointers, operator overloading, multiple inheritance, etc. String manipulations can be implemented very easily without any explicit concatenation procedure.
- **Portability:** Java is famous for its unique feature—platform independence (architecture neutral). This means that a Java program compiled on one machine could be ported to any other machine/operating system and executed without any modifications. Thus, the class file, which is the result of compilation resides on a DOS platform, can run on a UNIX platform unlike the .exe file of C/C++. This is an important feature for which Java is preferred for the Internet applications.
- **Robust and Object Oriented:** Java is a highly object oriented language where reusability is of utmost importance. Java programs are very reliable on different platforms with the special features of memory allocation and de-allocation, and exception handling. Memory management, especially de-allocation, is taken care of by the Java environment, which is not the case in C/C++, where programmers have to deal with it explicitly with extra code. Exception handling is a mechanism that helps the execution of a code, even if an error occurs at runtime, by handling the exception.

  Robustness is also achieved because Java is a strongly typed language. It signifies that you must declare the variable type before you use it. This is different from languages, such as PERL, JavaScript, etc., which are loosely typed.
- **Multithreaded:** Java with its multithreaded approach can run many programs concurrently, thereby saving processor time. Synchronization of code is an added feature of Java to run non-erroneous interactive applications.
- **Distributed and Dynamic:** Java is also popular for its distributed environment, as it supports the Transmission Control Protocol/Internet Protocol (TCP/IP). With Java, you can access a Uniform Resource Locator (URL) or a file on a remote server in some other country with the same ease, as you can access a file on your local system.

Java can also validate the code at runtime, which is more important for applets. Therefore, it is feasible to dynamically connect the code in a secure and practical manner.

NOTES

2.3 FEATURES OF JAVA

Java inherits most of its features from the earlier object oriented languages like C++. Various features of Java are:

- Compiled and interpreted.
- Platform independent and portable.
- Object oriented programming approach.
- Robust and secure.
- Simple and distributed.
- Small and familiar.
- Multithreaded and interactive.
- High performance.
- Dynamic and extensible.

Java Environment

The Java environment comprises a set of tools and classes that are used to run the Java program. These tools are:

- Java development kit.
- Java standard library.

Java Development Kit



Java Development Kit (JDK): It Contains various tools used by the Java Runtime Environment (JRE) that can be used to compile and interpret Java programs

Java Development Kit (JDK) contains various tools used by the Java Runtime Environment (JRE) that can be used to compile and interpret Java programs. The Java tools include Java compiler, Java interpreter, Java disassembler and Java debugger. JRE is a software that is used to execute the Java program. The Java Development Kit (JDK) contains following tools to run and execute the Java programs:

- `appletviewer` allows you to run Java applets.
- `javac` is a Java compiler that translates the Java source code into Javaclass files or the bytecode files that can be interpreted by the java interpreter.
- `java` is a Java interpreter that executes various applications by interpreting the bytecodes files.
- `javah` is used for including the C header files in a Java program.
- `javap` acts as a disassembler and is used to convert bytecode files into a program source code.
- `javadoc` creates documentation of the Java source code files in the HTML format from the comments available in the java source code.
- `jdb` is a java debugger that locates errors in a Java program.

Figure 2.1 shows the basic configuration of the Java tools that is used to process a Java program.

NOTES

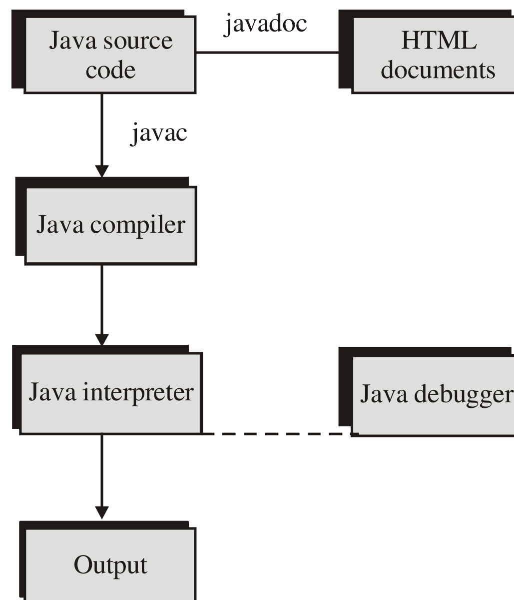


Fig. 2.1 *Implementation of the JDK Tools*

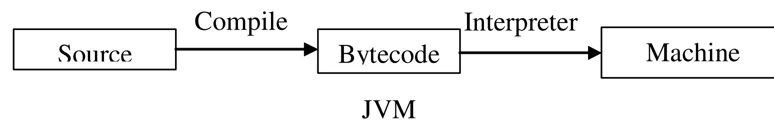
Java Standard Library

Java standard library is one of the most attractive features of Java that contain various classes to support all the major functions of Java. Java standard library includes the following classes:

- Language support classes which support basic features in Java, such as strings, arrays, threads and exception handling.
- Utility classes that are used for various functions, such as random number generator function, date and time function, and container classes function.
- Input or output classes that are used to read data from various input devices like keyboard and joysticks and write data on various output devices, such as printer and monitor.
- Networking classes which allow communication between different computers over a network.
- Abstract Window Toolkit (AWT) that is used for creating platform independent GUI applications.
- Applet that allows you to create applets which can be downloaded and run on a client browser.

Java Virtual Machine

Java uses both compiler and interpreter. The source code written in Java is compiled to generate bytecode and then this bytecode is interpreted to machine instructions for a specific machine. The bytecode generated by the compiler is not machine specific as shown in Figure 2.2. It is generated for a virtual machine known as JVM (Java Virtual Machine). It exists only inside the computer memory. This virtual machine is designed in such a way that it can be implemented on any existing processor and itself acts as a virtual processor chip. It hides the underlying operating system details from Java applications.



NOTES

Fig. 2.2 *Execution of a Java Program*

2.4 LEXICAL ISSUES

Java is the latest all purpose programming language. Its programs are collection of whitespace, identifiers, literals, comments, operators, separators and keywords.

- **Whitespace:** Whitespaces consist of space characters, tab characters and linefeed characters. The compiler removes all the whitespaces before compilation.
- **Identifiers:** Identifiers are names provided by the programmers. These names are assigned to functions, methods, classes, variables, and so on, to exclusively identify them to the compiler. Java identifiers can be of any length. However, there is an upper limit of 255 characters.

Identifiers could be any explanatory sequences of upper case and lower case letters, numbers or the underscore and dollar sign characters. They must begin with a letter or an undersign character (_) or a dollar sign ($). Subsequent characters could be alphabets (capital as well as small) and numbers 0 to 9; space characters or any other special character found on your keyboard is not allowed. The only other limitation is that Java keywords cannot be used as an identifier.

In Java, each character is represented by 2 bytes (16 bits). The 16-bit Unicode character can accommodate 2^{16} = 65536 different characters. This may cover all the alphabets in all the languages on this planet. Strictly speaking, any such character can be included in an identifier.

A few examples of **Valid** and **Invalid** Java Identifiers are as follows:

Valid	Invalid	
HelloWorld	Hello World	(Uses a space)
Hello_Mummy	Hello Mummy!	(Uses a space and punctuation mark)
heyDude3	3heyDude	(Begins with a numeral)
tall	short	(This is a Java keyword)
poundage	#age	(Does not begin with a letter)

Literals



Literals: They are elements and used in an invariant way in a program. They are also called constants and could be number, characters or strings

Literals are elements. They are used in an invariant way in a program. They are also called constants. Literals could be numbers (numeric literals), characters or strings.

Numeric literal might be an integer, a floating point number or a Boolean. Integers come in different formats depending on the base system used to represent that number. Decimal format (base 10), hexadecimal format (base 16) and octal format (base 8) are the three commonly used formats. Decimal numbers appear as ordinary number with a leading 0x, or 0X in a program (For example: int y=0×7B; or int Y=0×7B;). Octal numbers appear with a leading 0 (example: int y=0173;). Decimal numbers appear as 173=0×7B=0173.

NOTES

Floating point literals represent decimal numbers with a fractional part, such as 314.156. They can be represented in either standard or scientific notation. The numbers 314.156 can be expressed as 3.14156×10^2 in scientific notation. In a Java program one can write it as:

```
int y=314.156;
      or
int y=3.14156e2;
      or
int y=3.14156E2;
```

Boolean literals are treated as numbers because of the influence of C. In Java the keywords *True* and *False* are used to represent the Boolean states.

Integer	***Floating Point***	***Boolean***
100	98.6	True/False

Character literals represent 16-bit Unicode characters and appear within a pair of simple quotes in a program. For example, char ch = 'a'. Back slash (\) followed by the character code represents special characters. The latter are control characters and characters that cannot be printed.

The special characters that Java maintains are shown in the Table 2.1 that follows:

Table 2.1 *Special Characters Supported by Java*

Character Description	Character Representation
Back slash	`\ \`
Continuation	`\`
Back space	`\ b`
Carriage return	`\ r`
Form feed	`\ f`
Horizontal tab	`\ t`
New line	`\ n`
Single quote	`\ '`
Double quote	`\ "`
Unicode character	`\ udddd`
Octal number	`\ ddd`

String literals stand for many characters enclosed within a pair of double quotes: For example: "Hello World". In Java, string literals are implemented through a class called String class.

Comments

Comments are used in a program for documentation purpose. Comments give clarity to the source code. Java supports the following types of comments:

`/* Some comment */`	-This is the old C style comment.
`// Some other comment`	-This is the C++ comments.
`/** Another comment */`	-This is same as the C style comment with an extra benefit that it can be used with Java automatic documentation tool 'javadoc' to create automatic documentation from the source code.

NOTES

Separators

Separators are used to inform the Java compiler about how statements are grouped together in the code. Table 2.2 shows the separators supported by Java.

Table 2.2 *Separators Supported by Java*

Symbol	*Name*	*Purpose*
()	Parentheses	Used to contain lists of parameters in method definition and invocation.
{}	Braces	Used to contain the values of automatically initialized arrays.
[]	Brackets	Used to declare array types.
;	Semicolon	Terminates statements.
,	Comma	Separates consecutive identifiers in a variable declaration
.	Period	Used to separate package names from subpackages and classes.

Java Keywords

Java keywords are reserved words and have special meaning for Java compilers. They cannot be used as identifiers in a program. Java has a rich set of keywords, which are listed as follows:

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	Else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

2.5 STRAWBERRY PERL OR ACTIVE PERL

Strawberry Perl or **Active Perl** is a distribution of the Perl programming language for the Microsoft Windows platform. While most other distributions rely on the user having software development tools already set up to install certain Perl components, Strawberry Perl ships with the most commonly used tools preconfigured and packaged. It is a dramatic departure from other Perl distributions and has influenced other distributions to provide such development tools in their own distribution.

Strawberry Perl consists of:

- A Perl distribution (currently, Perl 5.16.2).
- A MinGW distribution, consisting of gcc, ld, gmake and other binutils. The binutils are a set of programming tools for creating and managing binary programs, object files, libraries, profile data and assembly source code originally written by programmers at Cygners Solutions. The binutils include the following commands:

Check Your Progress

1. What is Java?
2. Why Java is important to the Internet?
3. State any two features of Java.
4. What are the types of Java programs?
5. What are identifiers?

`as`	Assembler popularly known as GAS (Gnu ASsembler)
`ld`	Linker
`gprof`	Profiler
`addr2line`	Convert address to file and line
`ar`	Create, modify, and extract from archives
`c++filt`	Demangling filter for C++ symbols
`dlltool`	Creation of Windows dynamic-link libraries
`gold`	Alternative linker
`nlmconv`	Object file conversion to a NetWare Loadable Module
`nm`	List symbols in object files
`objcopy`	Copy object files, possibly making changes
`objdump`	Dump information about object files
`ranlib`	Generate indexes for archives
`readelf`	Display content of ELF files
`size`	List total and section sizes
`strings`	List printable strings
`strip`	Remove symbols from an object file
`windmc`	Generates Windows message resources
`windres`	Compiler for Windows resource files

- Windows installation scripts to allow Strawberry Perl to be easily uninstalled.
- Various non-standard but widely used Perl modules. These mostly relate to ease of installation of further extensions from the CPAN (Comprehensive Perl Archive Network) but also include various tools that enhance the ability to install packages from other sources such as the, Perl Active Toolkit, Perl Package Managers (PPMs) and the Perl Installation program.

ActivePerl allows installation of packages specially packaged for Windows, called PPMs, so users can easily install popular Perl modules. Since PPMs are prepackaged for Windows, they may be simpler to install than CPAN modules in Strawberry Perl. Some CPAN modules will not work on Windows, as they rely on Unixisms or operating-system-specific tools; relying on prepackaged PPMs which are known to work on Windows can avoid such problems.

2.6 OPERATORS

Java provides a rich operator environment. Most of the operators can be divided into arithmetic operators, bitwise operators, relational operators and logical operators.

Arithmetic Operators

Arithmetic operators are used in mathematical expressions in the same way that they are used in algebra. Table 2.5 lists the arithmetic operators.

NOTES



Arithmetic operators: Used in mathematical expressions in the same way that they are used in algebra

NOTES

Table 2.5 Arithmetic Operators

Java Operation	Arithmetic Operator	Java Expression
Addition	+	a + b
Subtraction	–	a – b
Multiplication	*	a * b
Division	/	a / b
Modulus	%	a % b
Increment	++	a ++
Decrement	--	a --
Addition assignment	+=	a + = b
Subtraction assignment	–=	a – = b
Multiplication assignment	*=	a * + b
Division assignment	/|	a / + b
Modulus assignment	%+	a % =b

The operands of the arithmetic operators can be either numeric or char type but not of Boolean type.

Bitwise Operators

Bitwise operators work on individual bits. As Java has the numeric data types as signed twos complements, great care should be taken when dealing with negative numbers, especially the shift operators sometimes give odd results with negative numbers. They are summarized in the Table 2.6.

Table 2.6 Bitwise Operators

Bitwise Operator	Java Operation	Java expression
~	Unary NOT	!a
&	AND	a & b
|	OR	a | b
^	Exclusive OR	a ^ b
>>	Right shift	a>>2 (right shifts by 2)
>>>	Right shift with zero fill	a>>>
<<	Left shift	a<<
&=	AND assignment	a & = b
|=	OR assignment	a | = b
^=	Exclusive OR assignment	a ^ = b
>>=	Shift right assignment	a >>= b
>>>=	Shift right zero fill assignment	a >>>= b
<<=	Left shift assignment	a <<= b

Shift operators are not commonly used, however, in image processing at pixel levels they are used extensively.

Unary Integer Operators

The unary integer operator acts on a single integer. Table 2.7 lists the unary integer operators.

NOTES

Table 2.7 Unary Integer Operators

Description	Operator
Increment	+ +
Decrement	– –
Negation	–
Bitwise complement	~

Binary Integer Operators

Binary integer operators act as a pair of integers. Table 2.8 shows all the binary integer operators.

Table 2.8 Binary Integer Operators

Description	Operator
Addition	+
Subtraction	–
Multiplication	*
Division	/
Modulus	%
Bitwise AND	&
Bitwise OR	|
Bitwise XOR	^
Left shift	<<
Right shift	>>
Zero-fill right shift	>>>

Relational Operators

These operators are used to test for equality of expressions. They result in a true or false. The relational operators are shown in Table 2.9.



Relational Operators: These operators are used to test for equality of expressions

Table 2.9 Relational Operators

Relational Operator	Java Operation	Java Expression
= =	Test for equality	a = = b
! =	Not equal to	a ! = b
<	Less than	a < b
>	Greater than	a > b
< =	Less than or equal to	a < = b
> =	Greater than or equal to	a > = b

Boolean Logical Operators

These operators work on logical values that are either true or false. The logical operators are shown in Table 2.10.

NOTES

Table 2.10 Boolean Logical Operators

Logical Operator	Java Operation	Java Expression
&	Logical AND	a & b
|	Logical OR	a | b
!=	Not equal to	a != b
^	Logical XOR	a ^ b
||	Short circuit OR	a || b
&&	Short circuit AND	a && b
&=	AND assignment	a &= b
|=	OR assignment	a |= b
!	Logical unary NOT	a ! b
^ =	XOR assignment	a ^= b
= =	Equal to	a == b
?:	Ternary if-then-else	a ? b : c

Assignment Operator

The assignment operator is the single equal sign, =. It works in Java in the same way in which it works in another language. The common form is:

```
var = expression;
```

Table 2.11 describes the various assignment operators in Java.

Table 2.11 Assignment Operator

Description	Operator
Simple assignment	=
Addition assignment	+=
Subtraction assignment	−=
Multiplication assignment	*=
Division assignment	/=
Modulus assignment	%=
AND assignment	&=
OR assignment	|=
XOR assignment	^=

2.6.1 The ? Operator

This operator is known as conditional operator. It is also referred to as ternary operator as it takes three items. The syntax of conditional operators is:

```
condition ? expression1 : expression2
```

If condition is true `expression1` will be evaluated. If condition is false `expression2` will be evaluated.

String Operators

Just like integers, floating point numbers strings can also be manipulated with operators. The only operation that is possible on string is the concatenation operation (+). Strings can be added together to get another string.

For example:

```
"Hello" + "World" = "Hello World"
```

NOTES

Program 1: A Java program to demonstrate string addition or string concatenation

```
public class StringAddition
{
   public static void main(String args[ ])
   {
      String s1="hello" =+ "World";
      String s2="Hello" + Beautiful" + "World";
      System.out.println(s1+s2);
      }
}
```

This program will print a single string output.

Output of the program:

```
Hello World Hello Beautiful World
```

Division by Zero

Java does not allow integer division by zero. The operation 10/0 is an error and the program will throw an ArithmeticException object. However, floating point division by zero is permitted. The operations 10.0/0, 10/0.0., 10.0/0.0, etc., are allowed. The result of this operation is infinity. The statement

```
System.out.println(10.0/0);
```

will print an output as follows:

```
Infinity
```

Operator Precedence

Operator precedence establishes the order in which operations are assessed. Each Java operator has a precedence connected with it. The table that follows shows the Java operators and their precedence. The operators at the topmost line have the highest precedence. It means in an expression these operators will be evaluated first. The operators in the bottommost line have the lowest precedence. Operators in the same line have the same priority. Operations of the same precedence are evaluated from left to right.

	[ ]	( )	
++	—	!	~
*	/	%	
+	-		
<<	>>	>>>	
= =	!=		
&			
^			
&& || ?} =			

2.7 CONTROL STATEMENTS

NOTES



Sequential execution: The type of execution in which the statements in a program are executed one by one in the order in which they are written

Generally, the statements in a program are executed one by one in the order in which they are written. This type of execution is called **sequential execution**. If a programmer wants to execute some statements even before other statements by surpassing the order, he must temporarily transfer the control of execution. This can be achieved with control statements like `if`, `for`, `switch`, etc. `goto` is a keyword in Java but its usage is prohibited keeping in view of the advantage of a structured programming language.

Selection Statements

The `if` statement is the simplest form of condition called a **single selection structure,** because it selects or ignores a single action. The syntax of `if` statement is:

```
if ( condition ) statement1;
else statement2;
int i, j;
// ….
if ( i < j ) i = 0;
else j = 0;
```

`if ... else ... else if` Statement

The `if ... else` structure is called a double selection structure, because it selects between two different actions. The syntax is as follows:

```
if (condition)
statement;
else if (condition)
statement;
else if (condition)
statement;
.
.
.
else
  statement;
```

Iteration Statements

Java's iterations statements are `for`, `while` and `do...while`. These statements create what is commonly called *loops.*

`for` Loop

In `for` loop, execution of a group of statements is repeated a designated number of times. The syntax of `for` loop is as follows:

```
for (initialization-expression;  test-condition;
        increment/decrement-operation)
 {
```

```
statement1;
statement2;
    .
    .
    .
    statement n;
    }
```

NOTES

The `for` loop repeats the group of statements {statement1,…, statementn;} a number of times that is decided by the initialization-expression, the loop test-condition and the increment/decrement-expression. Initialization-expression is used to initialize a loop control variable. The test condition tests the loop control variable for some limit value. Increment/decrement-expression specifies how the loop control variable has to be modified before the next iteration.

while Loop

Unlike the for loop, the while loop has no initialization expression or increment/decrement expression. It has a loop condition that controls the execution of the loop statement. The while loop is most fundamental loop statement in Java. It repeats a statement or block while its controlling expression is true. The syntax is:

```
While (loop condition)
{
statement1;
statement2;
           .
           .
           .
    statement n
}
```

If the loop condition evaluates to be true, then the statements {statement1; ….statementn} will be executed. If the loop condition evaluates to be false, the control will go to the first statement after the curly bracket enclosing the loop statements.

Another point to remember is that there is no increment/decrement expression. The execution of the body of the loop must alter the value of some variable to affect the loop condition. Otherwise, the loop statements will go on executing an infinite number of times.

The loop condition is at the beginning of the loop. It can happen that the loop condition is false at the beginning itself. In this case, the loop will not be executed at all. The while loop is suitable for iterations that takes place zero or more times, on the contrary, the for loop is suitable for iterations that take place a known number of times.

You can have a continue statement in a while loop as in the case of the for loop. When a continue statement is executed, the flow of control is transferred to the next iteration of the loop.

NOTES

`do ... while` Loop

The `do...while` loop is very much similar to the `while` loop. The syntax of the `do ... while` loop is as follows:

```
do
     {
statement1;
statement2;
          .
          .
          .
statement n;
} while(loop condition);
```

The loop condition is evaluated after the statements {`statement1;` ......`s;statementn;`} are executed once. If the loop condition evaluates to be true, then the statement {`statemnt1;`......`; statementn;`} will be executed again. If the loop condition evaluates to be false, the control will go to the first statement after the `while` statement.

Jump Statements

Java supports three jump statements: `break`, `continue` and `return.` These statements transfer control to another part of your program.

`break` Statement

The `break` statement is used to break from an enclosing `do ... while`, `for` or `switch` statement.

Program 2: A Java program to illustrate the `break` statement.

```
class Break
{
    public static void main(String args[])
            {
      for( int i= 0;i<5;i++)
           {
           if(i==2) break;
           System.out.println("number :" +i);
            }
        System.out.println("End of the for loop");
        }
    }
```

Output of the program:

```
Number: 0
Number:1
End of the for loop
```

`continue` Statement

The `continue` statement skips the current iteration of a `for`, `while` or `do ... while` loop.

NOTES

Program 3: A Java program to illustrate the `continue` statement.

```
class Continue
{
    public static void main(String args[])
            {
      for( int i= 0;i<5;i++)
           {
           if(i= =2) continue;
           System.out.println ("number:" +i);
            }
System.out.println("End of the for loop");
    }
}
```

Output of the program:

```
Number: 0
Number: 1
Number: 3
Number: 4
End of the for loop
```

`return` Statement

The last of the branching statements is the `return` statement. The `return` statement exits from the current method and control flow returns to where the method was invoked.

2.8 PRACTICAL EXTRACTION AND REPORTING LANGUAGE (PERL): AN INTRODUCTION

Perl was created by programmer Larry Wall. He wanted to have all the best features of other programming languages rolled into one and leave out all the useless functions. Common Gateway Interface (CGI) is the means by which a Perl program, executed on the Web server, communicates with a client (i.e., a user with a Web browser).

Learning how to program in Perl is an essential skill for the one who wish to create advanced, interactive Websites. On the Web these days, Perl powers many Web applications. Forms, search engines, guest books, electronic greetings, counters, automatic page generation, mailing lists and databases are all made possible through Perl and CGI. If you learn Perl and CGI, there are endless scope for you.

Perl is an interpreted programming language, as opposed to a compiled programming language. 'Interpreted' means that a program written in Perl must be executed by the Perl interpreter every time it is executed. In contrast, a program written in a language, such as C or C++ must be 'compiled' once and then can be directly executed later.

Check Your Progress

6. What is Strawberry Perl?
7. What are the constituents of Strawberry Perl?
8. List the operators used in Java.
9. Which control statements are used in Java?



Perl: It is an interpreted programming language, as opposed to a compiled programming language

NOTES

Since Perl is an interpreted language, it is quite easy to write a program, run and test it, modify it, run it again, and so on, because recompiling after each change is not necessary. However, this can be a limitation, as it requires that the Perl interpreter be present every time the program is run. Most WWW servers have Perl installed, so using Perl for CGI is usually possible.

2.8.1 Basic Structure of Perl

Perl is structurally similar to Pascal and C, so if you have got experience in these programming languages, Perl should be fairly simple to pick up. There are small differences that may cause problems if you are not attentive. A list of some of these differences can be found at respective Websites, along with advice for programmers coming to Perl from other languages.

Perl, like most languages, is made up of statements. These are instructions that manipulate data, control the program flow, and handle input and output. Typically, there is one statement per line. Each line in Perl must end with a semicolon. This is so that the interpreter can differentiate between commands. Whitespace (spaces, tabs, blank lines) is ignored and for this reason, use of whitespace to increase readability is recommended. Anything from a '`#`' character to the end of a line is treated as a comment and also ignored.

Note: There is no syntax in Perl for multi-line comments, like there is in C (`'/*'` and `'*/'`). Usage of `'#'` is equivalent to the C++ usage of `'//'`.

2.9 STARTING A PERL PROGRAM

Every Perl program begins with a line resembling '`#!/usr/bin/Perl`', where '`/usr/bin/Perl`' should be replaced with the actual path to the Perl interpreter on your system. On a system if the interpreter is located in `/usr/local/bin`, the first line should be '`#!/usr/local/bin/Perl`'. If you are not sure, consult your system administrator.

Perl Files

Perl files generally have the extension '.pl'. For CGI programs, the extension '.cgi' may be used in order to tell the WWW server software that the file is meant to be executed, rather than sent to the browser directly. On UNIX-like systems, make sure the file is set `+x`. This usually translates into `chmod 755`.

The CGI.pm Module

Perl offers a powerful feature to programmers, namely add-on modules. These are collections of pre-written code that you can use to do all kinds of tasks. You can save the time and trouble of reinventing the wheel by using these modules.

Some modules are included as part of the Perl distribution; these are called *standard library modules* and do not have to be installed. If you have Perl, you already have the standard library modules.

There are also many other modules available that are not part of the standard library. These are typically listed on the Comprehensive Perl Archive Network (CPAN), which you can search on the Web at http://search.cpan.org.

The CGI.pm module is part of the standard library and has been since Perl version 5.004. CGI.pm has a number of useful functions and features for writing CGI programs and its use is preferred by the Perl community.

Let us see how to use a module in your CGI program. First you have to actually include the module via the `use` command. This goes after the `#!/usr/bin/Perl` line and before any other code:

```
use CGI qw(:standard);
```

Note: We are not going to use CGI.pm but rather use CGI. The .pm is implied in the `use` statement. The `qw(:standard)` part of this line indicates that we are importing the 'standard' set of functions from CGI.pm.

Now you can call the various module functions by typing the function name followed by any arguments:

```
functionname(arguments)
```

If you are not passing any arguments to the function, you can omit the parentheses.

A **function** is a piece of code that performs a specific task; it may also be called a *subroutine* or a *method.* Functions may accept optional *arguments* (also called *parameters*), which are values (strings, numbers and other variables) passed into the function for it to use. The CGI.pm module has many functions. For now we will start by using these three:

```
header;
start_html;
end_html;
```

The `header` function prints out the 'Content-type' header. With no arguments, the type is assumed to be 'text/html' `start_html` prints out the `<HTML>`, `<HEAD>`, `<TITLE>` and `<BODY>` tags. It also accepts optional arguments. If you call `start_html` with only a single string argument, it is assumed to be the page title.

NOTES



Function: It is a piece of code that performs a specific task; it may also be called a subroutine or a method

Program 4: A program for printing an HTML file

```
print start_html("Hello World");
```

It will print out the following:

```
<HTML>
<HEAD>
<TITLE>Hello World</TITLE>
<HEAD>
<BODY>
```

You can also set the page colors and background image with `start_html`:

```
print start_html(-TITLE=>"Hello World",
    -BGCOLOR=>"#cccccc", -TEXT=>"#999999",
    -BACKGROUND=>"bgimage.jpg");
```

Note: With multiple arguments, you have to specify the name of each argument with `-TITLE=>`, `-BGCOLOR=>`, etc. This example generates the same HTML as above, only the body tag indicates the page colours and background image:

```
<BODY BGCOLOR="#CCCCCC" TEXT="#999999" BACKGROUND
="bgimg.jpg">
```

The `end_html` function prints out the closing HTML tags:

```
</BODY>
</HTML>
```

NOTES

2.9.1 Documenting Programs

Documentation can be embedded in a program using comments. A comment in Perl is preceded by the # sign, anything appearing after the # is a comment:

Program 5: A program 'Hello World' with comments

```
#!/usr/bin/Perl -wT
use CGI qw(:standard);
# This is a comment
# So is this
#
# Comments are useful for telling the reader
# what's happening. This is important if you
# write code that someone else will have to
# maintain later.
print header;   # here's a comment. print the header
print start_html("Hello World");
print "<h2>Hello, world!</h2>\n";
print end_html; # print the footer
# the end.
```

2.9.2 Debugging Programs

A number of problems can occur with your CGI programs and unfortunately the default response of the Web server when it encounters an error (the 'Internal Server Error') is not very useful for figuring out what happened.

If you see the code for the actual Perl program instead of the desired output page from your program, this probably means that your Web server is not properly configured to run CGI programs. You will need to ask your Webmaster how to run CGI programs on your server. And if you are the Webmaster, check your server's documentation to see how to enable CGI programs.

If you get an Internal Server Error, there is either a permissions problem with the file or a bug in your program. A good first step in debugging is to use the `CGI::Carp` module in your program:

```
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);
```

This causes all warnings and fatal error messages to be echoed in your browser window. You will want to remove this line after you are finished developing and debugging your programs, because `Carp` errors can give away important security info to potential hackers.

If you are using the `Carp` module and are still seeing the 'Internal Server Error', you can further test your program from the command line in the UNIX shell. This will check the syntax of your program without actually running it:

```
Perl -cwT fourth.cgi
```

If there are errors, it will report any syntax errors in your program:

```
% Perl -cwT fourth.cgi
syntax error at fourth.cgi line 5, near "print"
fourth.cgi had compilation errors.
```

This tells you that there is a problem on or around line 5, make sure you did not forget a closing semicolon on the previous line, and check for any other typos. Also be sure you saved and uploaded the file as text, hidden control characters or smart quotes can cause syntax errors, too.

Another way to get more information about the error is to look at the Web server log files. Usually this will show you the same information that the `CGI::Carp` module does, but it is good to know where the server logs are located and how to look at them. Some usual locations of sever logs are `/usr/local/etc/httpd/logs/error_log` or `/var/log/httpd/error_log`. Ask your ISP if you are not sure of the location. In the UNIX shell, you can use the `tail` command to view the end of the log file:

```
tail /var/log/apache/error_log
```

The last line of the file should be your error message (although if you are using a shared Web server like an ISP, there will be other users' errors in the file as well). Here are some example errors from the error log:

```
[Fri Jan 16 02:06:10 2004] access to /home/book/ch1/test.cgi
failed for 205.188.198.46, reason:
malformed header from script.
In string, @yahoo now must be written as \@yahoo at /home/
book/ch1/test.cgi line 331, near "@yahoo"
Execution of /home/book/ch1/test.cgi aborted due to
compilation errors.
[Fri Jan 16 10:04:31 2004] access to /home/book/ch1/test.cgi
failed for 204.87.75.235, reason: Premature end of script
headers
```

A '`malformed header`' or '`premature end of script headers`' can either mean that you printed something before printing the '`Content-type: text/html`' line or your program died. An error usually appears in the log indicating where the program died as well.

NOTES

Check Your Progress

10. Who developed Perl?
11. What is the first step in debugging?
12. How a comment is written in Perl?
13. What are the types of variable?
14. What are environment variable?

2.10 PERL VARIABLES

Before you can proceed much further with CGI programming, you will need some understanding of Perl variables and data types. A variable is a place to store a value, so you can refer to it or manipulate it throughout your program. Perl has three types of variables, namely (i) Scalars, (ii) Arrays and (iii) Hashes.

Scalars

A **scalar variable** stores a single (scalar) value. Perl scalar names are prefixed with a dollar sign (`$`). `$x`, `$y`, `$z`, `$username` and `$url` are all examples of scalar variable names. Here is how variables are set:

```
$foo = 1;
$name = "Fred";
$pi = 3.141592;
```



Scalar variable: It stores a single (scalar) value

NOTES

In this example `$foo, $name` and `$pi` are scalars. You do not have to declare a variable before using it, but it is considered good programming style to do so. There are several different ways to declare variables but the most common way is with the `my` function:

```
my $foo = 1;
my ($name) = "Fred";
my ($pi) = 3.141592;
```

`my` simultaneously declares the variables and limits their *scope* (the area of code that can see these variables) to the enclosing code block. You can declare a variable without giving it a value:

```
my $foo;
```

You can also declare several variables with the same `my` statement:

```
my ($foo, $bar, $blee);
```

You can omit the parentheses if you are declaring a single variable, however, a list of variables must be enclosed in parentheses.

A scalar can hold data of any type, be it a string or a number. You can also use scalars in double quoted strings:

```
my $fnord = 23;
my $blee = "The magic number is $fnord.";
```

Now if you print `$blee`, you will get '`The magic number is 23.`' Perl interpolates the variables in the string, replacing the variable name with the value of that variable.

Let us try it out in a CGI program. Start a new program called scalar.cgi:

Program 6: A program to print scalar variables

```
#!/usr/bin/Perl -wT
use CGI qw(:standard);
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);
use strict;
my $email = "fnord\@cgi101.com";
my $url = "http://www.cgi101.com";
print header;
print start_html("Scalars");
print <<EndHTML;
<H2>Hello</H2>
<P>
My e-mail address is $email, and my Web url is
<a href="$url">$url</A>.
</p>
EndHTML
print end_html;
```

Note: You may change the `$email` and `$url` variables to show your own e-mail address.

NOTES

You will also notice the variable declarations:

```
my $email = "fnord\@cgi101.com";
my $url = "http://www.cgi101.com";
```

Notice that the @ sign in the e-mail address is preceded by a backslash. This is because the @ sign means something special to Perl — just as the dollar sign indicates a scalar variable, the @ sign indicates an array, so if you want to actually use special characters like @, $ and % inside a double quoted string, you have to precede them with a backslash (\).

A better way to do this would be to use a single quoted string for the e-mail address:

```
my $email = 'fnord@cgi101.com';
```

Single quoted strings are not interpolated the way double quoted strings are, so you can freely use the special characters $, @ and % in them. However, this also means you cannot use a single quoted string to print out a variable, because:

```
print '$fnord';
```

It will print the actual string '$fnord' . . . not the value stored in the variable named $fnord.

Arrays

An array stores an ordered list of values. While a scalar variable can only store one value, an array can store many. Perl array names are prefixed with an @ sign. Here is an example:

```
my @colors = ("red","green","blue");
```

Each individual item (or element) of an array are referred by its index number. Array indices start with 0, so to access the first element of the array @colors you use $colors[0]. Notice that when you are referring to a single element of an array, you prefix the name with $ instead of @. The $ sign again indicates that it is a single (scalar) value, the @ sign means you are talking about the entire array.

If you want to loop through an array, printing out all of the values, you could print each element one at a time:

```
my @colors = ("red","green","blue");
print "$colors[0]\n";       # prints "red"
print "$colors[1]\n";       # prints "green"
print "$colors[2]\n";       # prints "blue"
```

A much easier way to do this is to use a foreach loop:

```
my @colors = ("red","green","blue");
foreach my $i (@colors)
{
    print "$i\n";
}
```

For each iteration of the foreach loop, $i is set to an element of the @colors array. In this example, $i is 'red' the first time through the loop. The braces {} define where the loop begins and ends, so for any code appearing between the braces, $i is set to the current loop iterator.

NOTES

Notice we have used `my` again here to declare the variables. In the `foreach` loop, `my $i` declares the loop iterator (`$i`) and also limits its scope to the `foreach` loop itself. After the loop completes, `$i` no longer exists.

Getting Data Into and Out of Arrays

An array is an ordered list of elements. You can think of it like a group of people standing in line waiting to buy tickets. Before the line forms, the array is empty:

```
my @people = ();
```

Then Krishan walks up. He is the first person in line. To add him to the `@people` array, use the `push` function:

```
push(@people, "Krishan");
```

Now Sukriti, Aditi, Jai get in line. Again they are added to the array using the `push` function. You can push a list of values onto the array:

```
push(@people, ("Sukriti", "Aditi", "Jai"));
```

This pushes the list containing *"Sukriti", "Aditi"* and *"Jai"* onto the end of the `@people` array, so that `@people` now looks like this: (*"Krishan", "Sukriti", "Aditi" , "Jai")*

Now the ticket office opens and Krishan buys his ticket and leaves the line. To remove the first item from the array, use the `shift` function:

```
my $who = shift(@people);
```

This sets `$who` to "`Krishan`" and also removes "`Krishan`" from the `@people` array. So `@people` now looks like this: (*"Sukriti", "Aditi", "Jai")*

Suppose `Jai` gets paged and has to leave. To remove the last item from the array, use the `pop` function:

```
my $who = pop(@people);
```

This sets `$who` to `"Jai"` and `@people` is now (*"Sukriti", "Aditi")*

Both `shift` and `pop` change the array itself by removing an element from the array.

Finding the Length of Arrays

If you want to find out how many elements are in a given array, you can use the `scalar` function:

```
my @people = ("Krishan", "Sukriti", "Aditi", "Jai")
my $linelen = scalar(@people);
print "There are $linelen people in line.\n";
```

This prints "`There are 4 people in line.`" Of course, there's always more than one way to do things in Perl and that's true here — the `scalar` function is not actually needed. All you have to do is evaluate the array in a `scalar` context. You can do this by assigning it to a `scalar` variable:

```
my $linelen = @people;
```

This sets `$linelen` to 4.

What if you want to print the name of the last person in line? Remember that Perl array indices start with 0, so the index of the last element in the array is actually of length`-1`:

```
print "The last person in line is $people[$linelen-1].\n";
```

Perl also has a handy shortcut for finding the index of the last element of an array, the `$#` shortcut:

```
print "The last person in line is $people[$#people].\n";
```

`$#arrayname` is equivalent to `scalar(@arrayname)-1`. This is often used in `foreach` loops where you loop through an array by its index number:

NOTES

```
my @colors = ("cyan", "magenta", "yellow", "black");
foreach my $i (0..$#colors)
{
  print "color $i is $colors[$i]\n";
}
```

This will print out "`color 0 is cyan, color 1 is magenta`", etc.

The `$#arrayname` syntax is one example where an `#` sign does not indicate a comment.

Array Slices

You can retrieve part of an array by specifying the range of indices to retrieve:

```
my @colors = ("cyan", "magenta", "yellow", "black");
my @slice = @colors[1..2];
```

This example sets `@slice` to `("magenta", "yellow").`

Finding an Item in an Array

If you want to find out if a particular element exists in an array, you can use the `grep` function:

```
my @results = grep(/pattern/,@listname);
```

`/pattern/` is a regular expression for the pattern you are looking for. It can be a plain string, such as `/Box kite/` or a complex regular expression pattern.

`/pattern/` will match partial strings inside each array element. To match the entire array element, use `/^pattern$/`, which anchors the pattern match to the beginning (`^`) and end `($)` of the string.

Sorting Arrays

You can do an alphabetical (ASCII) `sort` on an array of strings using the sort function:

```
my @colors = ("cyan", "magenta", "yellow", "black");
my @colors2 = sort(@colors);
```

`@colors2` becomes the `@colors` array in alphabetically sorted order ("`black`", "`cyan`", "`magenta`", "`yellow`"). Note that the `sort` function, unlike `push` and `pop`, does not change the original array. If you want to save the sorted array, you have to assign it to a variable. If you want to save it back to the original array variable, you would do:

```
@colors = sort @colors;
```

You can invert the order of the array with the reverse function:

```
my @colors = ("cyan", "magenta", "yellow", "black");
@colors = reverse(@colors);
@colors is now ("black", "yellow", "magenta", "cyan").
```

NOTES

To do a reverse sort, use both functions:

```
my @colors = ("cyan", "magenta", "yellow", "black");
@colors = reverse(sort(@colors));
@colors is now ("yellow", "magenta", "cyan", "black").
```

Joining Array Elements into a String

You can merge an array into a single string using the `join` function:

```
my @colors = ("cyan", "magenta", "yellow", "black");
my $colorstring = join(", ",@colors);
```

This joins `@colors` into a single string variable (`$colorstring`), with each element of the `@colors` array combined, and separated by a comma and a space. In this example `$colorstring` becomes '`cyan`, `magenta`, `yellow`, `black`'.

You can use any string (including the empty string) as the separator. The separator is the first argument to the `join` function:

```
join(separator, list);
```

The opposite of `join` is `split`, which splits a string into a list of values.

2.11 CGI ENVIRONMENT VARIABLES

Environment variables are a series of hidden values that the Web server sends to every CGI program you run. Your program can parse them and use the data they send. Environment variables are stored in a hash named `%ENV`(refer Table 2.12).

Table 2.12 *Environment Variables*

Key	Value
DOCUMENT_ROOT	The root directory of your server.
HTTP_COOKIE	The visitor's cookie, if one is set.
HTTP_HOST	The hostname of the page being attempted.
HTTP_REFERER	The URL of the page that called your program.
HTTP_USER_AGENT	The browser type of the visitor.
HTTPS	'on' if the program is being called through a secure server.
PATH	The system path your server is running under.
QUERY_STRING	The query string.
REMOTE_ADDR	The IP address of the visitor.
REMOTE_HOST	The hostname of the visitor (if your server has reverse-name-lookups on; otherwise this is the IP address again).
REMOTE_PORT	The port the visitor is connected to on the Web server.
REMOTE_USER	The visitor's username (for .htaccess protected pages).
REQUEST_METHOD	GET or POST
REQUEST_URI	The interpreted pathname of the requested document or CGI (relative to the document root).
SCRIPT_FILENAME	The full pathname of the current CGI.
SCRIPT_NAME	The interpreted pathname of the current CGI (relative to the document root).
SERVER_ADMIN	The e-mail address for your server's Webmaster.
SERVER_NAME	Your server's fully qualified domain name (e.g., www.cgi101.com).
SERVER_PORT	The port number your server is listening on.
SERVER_SOFTWARE	The server software you are using (e.g., Apache 1.3).

Some servers set other environment variables as well, check your server documentation for more information. Notice that some environment variables give information about your server and will never change (such as, `SERVER_NAME` and `SERVER_ADMIN`), while others give information about the visitor and will be different every time someone accesses the program.

Not all environment variables get set. `REMOTE_USER` is only set for pages in a directory or subdirectory that is password protected via a .htaccess file. Even then, `REMOTE_USER` will be the username as it appears in the .htaccess file; it is not the person's e-mail address. There is no reliable way to get a person's e-mail address, short of asking them for it with a Web form.

You can print the environment variables the same way you would any hash value:

```
print "Caller = $ENV{HTTP_REFERER}\n";
```

Let us try printing some environment variables. Start a new file named env.cgi:

Program 7: A program to print environment variables

```
#!/usr/bin/Perl -wT
use strict;
use CGI qw(:standard);
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);
print header;
print start_html("Environment");
foreach my $key (sort(keys(%ENV)))
{
    print "$key = $ENV{$key}<br>\n";
}
print end_html;
```

Referring Page

When you click on a hyperlink on a Web page, you are being referred to another page. The Web server for the receiving page keeps track of the referring page and you can access the URL for that page via the `HTTP_REFERER` environment variable.

Program 8: A program that illustrates HTTP referrer

```
#!/usr/bin/Perl -wT
use CGI qw(:standard);
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);
use strict;
print header;
print start_html("Referring Page");
print "Welcome, I see you've just come from
$ENV{HTTP_REFERER}!<p>\n";
print end_html;
```

NOTES

Check Your Progress

15. Fill in the blanks with appropriate words.
    (a) The Java ___________ comprises a set of tools and classes that are used to run the Java program. These tools are:
    (b) The only operation that is possible on string is the ______________ operation.
    (c) ________________ means that a program written in Perl must be executed by the Perl interpreter every time it is executed.
    (d) __________________ is only set for pages in a directory or subdirectory that is password protected via .ht access file.
16. State whether the following statements are true or false.
    (a) A prefix increment/ decrement operation takes effect prior to the evaluation of the expression in which it is appearing.
    (b) Every integer literal in a program does not have a default data type associated with it.
    (c) A scalar can hold data of any type, be it a string or a number.
    (d) The key `SERVER_PORT` has the value as the port the visitor is connected to on the Web server.

NOTES

2.12 SUMMARY

- Java is an object oriented programming language based on the concept of an object. Java is derived from C++, but is slightly simplified with libraries convenient for the Internet. It is the programming language for the Internet environment.
- The Internet implies heterogeneous systems, different network features, different windows libraries and different operating systems. Java guarantees identical program behavior on different platforms.
- Java is a highly object oriented language where reusability is of utmost importance. Java programs are very reliable on different platforms with the special features of memory allocation and de-allocation and exception handling.
- The syntax for the Java language is similar to that of the C++ and C. The end user, who is familiar with C and C++, requires less effort in learning Java because Java and C or C++ has similar syntax.
- The Java environment comprises a set of tools and classes that are used to run the Java program.
- Java Development Kit (JDK) contains various tools used by the Java Runtime Environment (JRE) that can be used to compile and interpret Java programs.
- Java standard library is one of the most attractive features of Java that contains various classes to support all the major functions of Java.
- Identifiers could be any explanatory sequences of upper case and lower case letters, numbers or the underscore and dollar sign characters.
- Literals are elements. They are used in an invariant way in a program. They are also called constants. Literals could be numbers (numeric literals), characters or strings.
- Strawberry Perl or Active Perl is a distribution of the Perl programming language for the Microsoft Windows platform. It is a dramatic departure from other Perl distributions and has influenced other distributions to provide such development tools in their own distribution.
- While most other distributions rely on the user having software development tools already set up to install certain Perl components, Strawberry Perl ships with the most commonly used tools preconfigured and packaged.
- ActivePerl allows installation of packages specially packaged for Windows, called PPMs, so users can easily install popular Perl modules.
- Java provides a rich operator environment. Most of the operators can be divided into arithmetic operators, relational operators and logical operators. Arithmetic operators are used in mathematical expressions in the same way that they are used in algebra.
- Operator precedence establishes the order in which operations are assessed. Each Java operator has a precedence connected with it.
- The statements in a program are executed one by one in the order in which they are written. This type of execution is called sequential execution.
- Common Gateway Interface (CGI) is the means by which a Perl program, executed on the Web server, communicates with a client (i.e., a user with a Web browser).

NOTES

- Perl is an interpreted programming language, as opposed to a compiled programming language. 'Interpreted' means that a program written in Perl must be executed by the Perl interpreter every time it is executed.
- Every Perl program begins with a line resembling '#!/usr/bin/Perl', where '/usr/bin/Perl' should be replaced with the actual path to the Perl interpreter on your system.
- Perl offers a powerful feature to programmers, namely add-on modules. These are collections of pre-written code that you can use to do all kinds of tasks.
- An array stores an ordered list of values. While a scalar variable can only store one value, an array can store many. Perl array names are prefixed with an `@` sign.
- Each individual item (or element) of an array may be referred to by its index number. Array indices start with 0, so to access the first element of the array `@colors` you use `$colors[0]`.
- Environment variables are a series of hidden values that the Web server sends to every CGI program you run. Your program can parse them and use the data they send. Environment variables are stored in a hash named `%ENV`.
- There are two ways to send data from a Web form to a CGI program, i.e., `GET` and `POST`. These methods determine how the form data is sent to the server.

2.13 ANSWERS TO 'CHECK YOUR PROGRESS'

1. Java is an object oriented programming language that is based on the concept of an object.
2. Java came to the forefront of programming with the help of the Internet. Consequently, Java had a deep effect on the Internet.
3. Various features of Java are:
    - Compiled and interpreted.
    - Platform independent and portable.
4. Java programs are of two types—applications and applets.
    (a) A Java application is a standalone program that can be invoked from the command line.
    (b) A Java applet is a small program embedded in a Web page, and it will be run when that page is browsed using a Web browser.
5. Identifiers are names provided by the programmers. These names are assigned to functions, methods, classes, variables, and so on, to exclusively identify them to the compiler. Java identifiers can be of any length. However, there is an upper limit of 255 characters.
6. Strawberry Perl or Active Perl is a distribution of the Perl programming language for the Microsoft Windows platform.
7. Strawberry Perl consists of the following:
    - A Perl distribution (currently, Perl 5.16.2).
    - A MinGW distribution, consisting of gcc, ld, gmake and other binutils.
    - Windows installation scripts.
    - Perl models.

NOTES

8. The various operators used in Java are:
   - Arithmetic operators.
   - Bitwise operators.
   - Relational operators.
9. The various control statements used in Java are:
   - Selection statements.
   - Iterative statements.
   - Jump statements.
10. Perl was created by programmer Larry Wall.
11. A good first step in debugging is to use the `CGI::Carp` module in your program.
12. A comment in Perl is preceded by the # sign; anything app earing after the # is a comment.
13. Perl has three types of variables, namely scalars, arrays and hashes.
14. Environment variables are a series of hidden values that the Web server sends to every CGI program you run. Your program can parse them and use the data they send.
15. (a) Environment, (b) Concatenation, (c) Interpreted, (d) `REMOTE_USER`.
16. (a) True, (b) False, (c) True, (d) False.

2.14 QUESTIONS AND EXERCISES

Short-Answer Questions

1. List the various Java keywords.
2. Distinguish between valid and invalid Java identifiers.
3. What is a Java token set?
4. Write some of the important features of Java programming language.
5. What are the components of Java development kit?
6. What is the difference between Java variables and Java literals?
7. What is the basic structure of Perl?
8. Name the add-on modules of Perl.
9. What are CGI environment variables?

Long-Answer Questions

1. Explain the different data types used in Java.
2. Discuss the significance of typecasting and type conversion.
3. Explain the various Java operators and operator precedence.
4. What are the various control statements? Explain each with their syntax.
5. Explain the automatic type promotion in expressions with the help of example.
6. Explain the basic structure of Perl.
7. Discuss the add-on modules of Perl.
8. Explain CGI environment variables with the help of examples.

UNIT 3 INTERNET SCRIPTING LANGUAGES

NOTES

Structure

3.0 INTRODUCTION

In this unit, you will learn about the basics of the Internet scripting languages, such as JavaScript and VBScript. JavaScript is an object oriented scripting language which enables programmatic access to objects within the client application and other applications. 'JavaScript' is a trademark of Sun Microsystems. It was used under license for technology invented and implemented by Netscape Communications and current entities, such as the Mozilla Foundation. The primary use of JavaScript is to write functions that are embedded or included from HTML pages and interact with the Document Object Model (DOM) of the page. VBScript is a scripting language or more specifically a scripting environment that can enhance HTML Web pages and makes them more active than a simple static display.

NOTES

You will learn about applets which is a small Java program and is much more focused in scope than a full-fledged Java application. Applets are not given access to a local machine's file system. Thus, applet can neither read a file from the local machine, nor it can save a file on the local system. The Applet class, which is in the java.awt.applet package, inherits the properties of the Panel class, which is in java.awt package. Finally, you will learn about Remote Method Invocation (RMI) which is a part of Java Development Kit (JDK). It allows to develop distributed application. Distributed systems require computations that are running in different address spaces, particularly on different machines, to be able to communicate between one another.

3.1 UNIT OBJECTIVES

After going through this unit, you should be able to:

- Understand the significance of JavaScript
- Learn about the basics VBScript
- Analyse various other scripting languages
- Describe the functionality of applets
- Explain the significance of socket programming
- Understand how to look up Internet address
- Learn how to retrieve Web pages
- Explain the significance of remote method invocation

3.2 JAVASCRIPT



JavaScript: It is the scripting language of the Web. JavaScript is used in millions of Web pages to add functionality, validate forms, detect browsers, and much more

JavaScript is the scripting language of the Web. JavaScript is used in millions of Web pages to add functionality, validate forms, detect browsers, and much more.

JavaScript is an object oriented scripting language which enables programmatic access to objects within both the client application and other applications. It is primarily used in the form of client-side JavaScript, implemented as an integrated component of the Web browser, allowing the development of enhanced user interfaces and dynamic Websites. JavaScript is a dialect of the ECMAScript standard and is characterized as a dynamic, weakly typed, prototype-based language with first-class functions. JavaScript was influenced by many languages and was designed to look like Java, but to be easier for non-programmers to work with.

JavaScript was originally developed by Brendan Eich of Netscape under the name *Mocha*, which was later renamed to *Live Script*, and finally to JavaScript. The change of name from LiveScript to JavaScript roughly coincided with Netscape adding support for Java technology in its Netscape Navigator Web browser. JavaScript was first introduced and deployed in the Netscape browser version 2.0B3 in December 1995. JavaScript, despite the name, is essentially unrelated to the Java programming language even though the two do have superficial similarities. Both languages use syntaxes influenced by that of C syntax, and JavaScript copies many Java names and naming conventions. The language's name is the result of a co-marketing deal between Netscape and Sun, in exchange for Netscape bundling Sun's Java runtime with their

then-dominant browser. The key design principles within JavaScript are inherited from the Self and Scheme programming languages.

NOTES

How Script Works?

Use in Web Pages: The primary use of JavaScript is to write functions that are embedded in or included from HTML pages and interact with the Document Object Model (DOM) of the page. Some simple examples of this usage are:

- Opening or popping up a new window with programmatic control over the size, position, and attributes of the new window, i.e., whether the menus, toolbars, etc. are visible.
- Validation of Web form input values to make sure that they will be accepted before they are submitted to the server.
- Changing images as the mouse cursor moves over them. This effect is often used to draw the user's attention to important links displayed as graphical elements.

Because JavaScript code can run locally in a user's browser (rather than on a remote server) it can respond to user actions quickly, making an application feel more responsive. Furthermore, JavaScript code can detect user actions which HTML alone cannot, such as individual keystrokes. Applications, such as Gmail take advantage of this. The user-interface logic is written in JavaScript and JavaScript dispatches requests for information (such as the content of an e-mail message) to the server. A JavaScript engine (also known as *JavaScript interpreter* or *JavaScript implementation*) is an interpreter that interprets JavaScript source code and executes the script accordingly. A Web browser is by far the most common host environment for JavaScript. Web browsers typically use the public API to create "host objects" responsible for reflecting the DOM into JavaScript. The Web server is another common application of the engine. A JavaScript Webserver would expose host objects representing an HTTP request and response objects, which a JavaScript program could then manipulate to dynamically generate Web pages.

A minimal example of a standards-conforming Web page containing JavaScript (using HTML 4.01 syntax) would be:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head><title>simple page</title></head>
  <body>
    <script type="application/javascript">
      document.write('Hello World!');
     </script>
     <noscript>
       <p>Your browser either does not support JavaScript,
or you have JavaScript turned off.</p>
     </noscript>
  </body>
</html>
```

Since JavaScript is the only language that the most popular browsers share support for, it has become a target language for many frameworks in other languages, even though JavaScript was never intended to be such a language. Despite the performance limitations inherent to its dynamic nature, the increasing speed of JavaScript engines have made the language a surprisingly feasible compilation target.

Uses Outside Web Pages

Outside the Web, JavaScript interpreters are embedded in a number of tools. Each of these applications provides its own object model which provides access to the host environment, with the core JavaScript language remaining mostly the same in each application.

- ActionScript, the programming language used in Adobe Flash, is another implementation of the ECMAScript standard.
- Apple's Dashboard Widgets, Microsoft's Gadgets, Yahoo! Widgets, Google Desktop Gadgets, Serence Klipfolio are implemented using JavaScript.
- The Mozilla platform, which underlies Firefox and some other Web browsers, uses JavaScript to implement the Graphical User Interface (GUI) of its various products.
- Adobe's Acrobat and Adobe Reader (formerly Acrobat Reader) support JavaScript in PDF files.
- Tools in the Adobe Creative Suite, including Photoshop, Illustrator, Dreamweaver and InDesign, allow scripting through JavaScript.
- The Qt C++ toolkit includes a QtScript module to interpret JavaScript, analogous to javax.script.
- OpenOffice.org office application suite allows for JavaScript as one of its scripting languages.
- Adobe Integrated Runtime is a JavaScript runtime that allows developers to create desktop applications.
- Some high-end Philips universal remote panels, including TSU9600 and TSU9400, can be scripted using JavaScript.
- Sphere is an open source and cross platform computer program designed primarily to make role-playing games that use JavaScript as a scripting language.
- The open-source Re-Animator framework allows developing 2D sprite-based games using JavaScript and XML.
- Methabot is a Web crawler that uses JavaScript as scripting language for custom filetype parsers and data extraction using E4X.
- Unity game engine supports three scripting languages: JavaScript, C#, and Boo.
- DX Studio (3D engine) uses the SpiderMonkey implementation of JavaScript for game and simulation logic.

Debugging

Within JavaScript, access to a debugger becomes invaluable when developing large, non-trivial programs. Because there can be implementation differences between the various browsers (particularly within the Document Object Model) it is useful to have access to a debugger for each of the browsers that a Web application targets. Script debuggers are available for Internet Explorer, Firefox, Safari, Google Chrome and Opera.

NOTES

Three debuggers are available for Internet Explorer: Microsoft Visual Studio is the richest of the three, closely followed by Microsoft Script Editor (a component of Microsoft Office), and finally the free Microsoft Script Debugger which is far more basic than the other two. The free Microsoft Visual Web Developer Express provides a limited version of the JavaScript debugging functionality in Microsoft Visual Studio.

Web applications within Firefox can be debugged using the Firebug add-on, or the older Venkman debugger. Firefox also has a simpler built-in Error Console, which logs and evaluates JavaScript. It also logs CSS errors and warnings. Opera includes a richer set of tools called DragonFly. WebKit's Web Inspector includes a JavaScript debugger in Apple's Safari. Some debugging aids are themselves bits of JavaScript code built to run on the Web. Web development bookmarklets and Firebug Lite provide variations on the idea of the cross-browser JavaScript console. Since JavaScript is interpreted, loosely-typed, and may be hosted in varying environments, each incompatible with the others, a programmer has to take extra care to make sure the code executes as expected in as wide a range of circumstances as possible and that functionality degrades gracefully when it does not.

What can a JavaScript do?

- **JavaScript gives HTML Designers a Programming Tool -** HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax. Almost anyone can put small 'snippets' of code into their HTML pages.
- **JavaScript can put Dynamic Text into an HTML Page -** A JavaScript statement like this: `document.write("<h1>" + name + "</h1>")` can write a variable text into an HTML page.
- **JavaScript can React to Events -** A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element.
- **JavaScript can Read and Write HTML Elements -** A JavaScript can read and change the content of an HTML element.
- **JavaScript can be Used to Validate Data -** A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing.
- **JavaScript can be Used to Detect the Visitor's Browser** - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser.
- **JavaScript can be Used to Create Cookies** - A JavaScript can be used to store and retrieve information on the visitor's computer.

JavaScript and Java

A common misconception is that JavaScript is similar or closely related to Java; this is not so. Both have a C-like syntax, are object oriented, are typically sandboxed and are widely used in client-side Web applications, but the similarities end there. Java has static typing; JavaScript's typing is dynamic (meaning a variable can hold an object of any type and cannot be restricted). Java is loaded from compiled bytecode; JavaScript is loaded as human-readable code. C is their last common ancestor language.

NOTES

Nonetheless, JavaScript was designed with Java's syntax and standard library in mind. In particular, all Java keywords are reserved in JavaScript, JavaScript's standard library follows Java's naming conventions, and JavaScript's Math and Date classes are based on those from Java 1.0.

The following is the simple example of a JavaScript program:

Program 1: A program to demonists ate `<script>` tag

```
<html>
<body>

<script type="text/javascript">
document.write("This is my first JavaScript!");
</script>

</body>
</html>
```

Output of the program:

This is my first JavaScript!

The HTML `<script>` tag is used to insert a JavaScript into an HTML page.

Put a JavaScript into an HTML Page

Program 2: A program shows how to use JavaSript to write text on a Web page:

```
<html>
<body>
<script type="text/javascript">
document.write("Hello World!");
</script>
</body>
</html>
```

Output of the program:

Hello World!

Scripts in `<head>` *and* `<body>`

You can place an unlimited number of scripts in your document, so you can have scripts in both the body and the head section.

```
<html>
<head>
<script type="text/javascript">
....
</script>
</head>
<body>
<script type="text/javascript">
....
</script>
</body>
```

JavaScript Statements: JavaScript is a sequence of statements to be executed by the browser. A JavaScript statement is a command to a browser. The purpose of the command is to tell the browser what to do. This JavaScript statement tells the browser to write "Hello Dolly" to the Web page:

```
document.write("Hello Dolly");
```

NOTES

It is normal to add a semicolon at the end of each executable statement. Most people think this is a good programming practice, and most often you will see this in JavaScript examples on the Web. The semicolon is optional (according to the JavaScript standard), and the browser is supposed to interpret the end of the line as the end of the statement. Because of this you will often see examples without the semicolon at the end.

Note: Using semicolons makes it possible to write multiple statements on one line.

JavaScript is Case Sensitive: Unlike HTML, JavaScript is case sensitive, therefore watch your capitalization closely when you write JavaScript statements, create or call variables, objects and functions.

JavaScript Blocks

JavaScript statements can be grouped together in blocks. Blocks start with a left curly bracket {, and ends with a right curly bracket }. The purpose of a block is to make the sequence of statements execute together. This example will write a heading and two paragraphs to a Web page:

Example

```
<script type="text/javascript">{document.write("<h1>This is a
heading</h1>");document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");}</script>
```

Output of the program:

This is a heading

This is a paragraph.

This is another paragraph.

The example above is not very useful. It just demonstrates the use of a block. Normally a block is used to group statements together in a function or in a condition (where a group of statements should be executed if a condition is met).

JavaScript Comments

JavaScript comments can be used to make the code more readable. Comments can be added to explain the JavaScript, or to make the code more readable. Single line comments start with //.

3.3 VBSCRIPT: AN INTRODUCTION

VBScript is acronym of **V**isual **B**asic **Script**ing Edition and is referred as an Active Scripting language specifically developed by Microsoft modeled on Visual Basic. This language is designed with a high-speed interpreter to be used with broad variety of Microsoft environments. VBScript uses the Component Object Model (COM) for accessing elements of the Microsoft environment within which it is running.



VBScript: A scripting language or more specifically a scripting environment that can enhance HTML Web pages and makes them more active than a simple static display.

NOTES

Because VBScript is an active scripting language hence it brings active scripting to a wide variety of environments, including Web client scripting in Microsoft Internet Explorer and Web server scripting in Microsoft Internet Information Service (IIS). The basic concepts of VBScript are common to most programming languages.

VBScript interacts with the host applications using Windows script. With Windows script, browsers and other host applications do not require special integration code for each scripting component. Windows script enables a host to compile scripts, obtain and call entry points and manage the namespace available to the developer.

VBScript can also be used to create applications that run directly on a workstation running Microsoft Windows. The simplest example is a script that makes use of the Windows Script Host (WSH) environment. Such a script is usually in a standalone file with the file extension .vbs. The script can be invoked in two ways. Wscript.exe is used to display output and receive input through a GUI, such as dialog and input boxes. Cscript.exe is used in a command line environment.

Since the language of VBScript is modelled on Visual Basic hence it can be analysed using the analogous categories, such as procedures, control structures, constants, variables, user interaction, array handling, date/time functions, error handling, mathematical functions, objects, regular expressions, string manipulation, etc.

In VBScript, a 'procedure' is the most important construct and is used to separate code into smaller modules. The function in VBScript returns a result in an assignment statement. Parameters are positional and can be passed by value or by reference. Control structures consist of the standard iterative and conditional `Do` loops, `If-Then-Else` and `Case` statements. Besides these, there are some additional complex variants, such as `ElseIf` and nested control structures. There are a large number of constants, such as 'True' and 'False' for logical values, `vbOKCancel` and `vbYesNo` for `MsgBox` codes, `vbBlack` and `vbYellow` for color values, `vbCR` for the carriage return character, etc.

By default, a variable is a well-located placeholder and refers to the memory location in a computer where the program information is stored or can be stored which may change when the script is running. In VBScript, constantly the variables are of one fundamental type, i.e., 'Variant' type. The user interaction is possible using the functions MsgBox and InputBox which display a simple dialog box format for messages and input. VBScript can be used in combination with HTML, for example, in an HTML Application.

The code modules are supported through the `<script>` and `</script>` tag. Each script section forms an independent code module that may have its own variables, functions and subroutines. The forms are created using the `<form>` and `</form>` tag and they are not visible as separate windows in the application. The forms are ways to group controls together for the purpose of addressing their properties and methods in code or to submit data to back-end process.

3.3.1 VBScript in HTML

VBScript is a scripting language or more specifically a scripting environment that can enhance HTML Web pages and makes them more active than a simple static display. VBScript is inserted into HTML, between standard `<script>` and `</script>` tags. Use the type attribute in the `<script>` tag to define the scripting language '`text/vbscript`':

NOTES

```
<html>
<body>
<script type="text/vbscript">
...
</script>
</body>
</html>
```

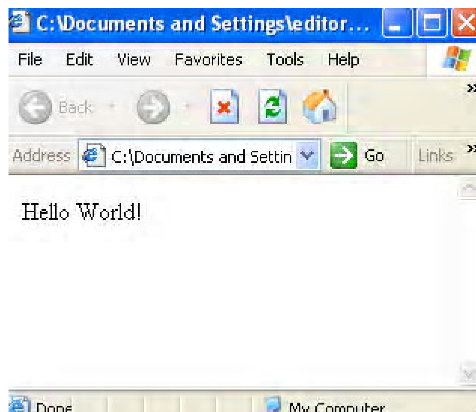
Internet explorer will interpret and execute the VBScript code between the `<script>` and `</script>` tags. As already mentioned VBScript is not a standard HTML language. VBScript is used in the Internet explorer only for learning purpose.

When VBScript is used on a Web server in an ASP page, the statement **`response.write()`** produces output. When we use the Internet explorer for testing VBScript, we use **`document.write()`** to produce output.

Program 3: A program that shows the use of `document write()`

```
<html>
<body>
<script type="text/vbscript">
document.write("Hello World!")
</script>
</body>
</html>
```

Output of the program:



Variables

In VBScript, variables are used to hold values or expressions. Variables are named storage locations that can contain data that can be modified while script is running. A variable can have a short name, like x or a more descriptive name, like car name or City. Rules for VBScript variable names are as follows:

- Must begin with a letter, i.e., an alphabetic character.
- Cannot contain a period (.).
- Cannot exceed 255 characters.
- Must be unique in the scope in which it is declared.

In VBScript, all variables are of type variant that can store different types of data.

NOTES

3.3.2 VBScript Data Types

VBScript has only one data type known as variant. A variant is a special kind of data type that can contain different kinds of information, depending on how it is used. Because variant is the only data type in VBScript, it is also the data type returned by all functions in VBScript. At its simplest, a variant can contain either numeric or string information.

Variant Subtypes

Beyond the simple numeric or string classifications, a variant can have other types as subtypes. For example, you can have numeric information that represents a date or a time. When used with other date or time data, the result is always expressed as a date or a time. These different categories of information that can be contained in a variant are called subtypes. Table 3.1 shows subtypes of data that a variant can contain.

Table 3.1 *Subtypes of Data that a Variant can Contain*

Subtype	Description
`Empty`	Variant is uninitialized. Value is 0 for numeric variables or a zero-length string ("") for string variables.
`Null`	Variant intentionally contains no valid data.
`Boolean`	Contains either True or False.
`Byte`	Contains integer in the range 0 to 255.
`Integer`	Contains integer in the range -32,768 to 32,767.
`Currency`	-922,337,203,685,477.5808 to 922,337,203,685,477.5807.
`Long`	Contains integer in the range -2,147,483,648 to 2,147,483,647.
`Single`	Contains a single precision, floating-point number in the range -3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values.
`Double`	Contains a double precision, floating point number in the range -1.79769313486232E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values.
`Date(Time)`	Contains a number that represents a date between January 1, 100 to December 31, 9999.
`String`	Contains a variable length string that can be up to approximately 2 billion characters in length.
`Object`	Contains an object.
`Error`	Contains an error number.

3.3.3 Functions in VBScript

A `Function` procedure is a series of statements, enclosed by the `Function` and `End Function` statements. It can perform actions and can return a value. It can take arguments that are passed to it by a calling procedure.

Syntax of a `Function` is:

```
Function myfunction()
  some statements
  myfunction=some value
End Function
```

NOTES

Or, alternatively use the following syntax:

```
Function myfunction(argument1,argument2)
  some statements
  myfunction=some value
End Function
```

3.4 OTHER SCRIPTING LANGUAGES

The following are significant scripting languages:

Client-Side Scripting Language

A scripting language or script language is referred as a programming language that supports in writing the program scripts. Scripts can be written and executed without explicit compile and link steps since they are typically created or modified by the user executing them. Characteristically, a scripting language is interpreted from source code or bytecode. On the contrary, in the software environment the scripts are typically written for a compiled language and distributed in machine code form where the user may not have access to its source code. The range of scripting languages varies from very small and highly domain specific programming languages to general purpose programming languages. The term script is typically used for small programs, i.e., programs up to a few thousand lines of code. Basically, the scripting languages occurred because of the development of the Internet as a communications tool. JavaScript, ASP (Active Server Pages), JSP (Java Server Pages), PHP (Hypertext Preprocessor), Perl, Tcl and Python are examples of scripting languages.

Client-side scripts are embedded within an HTML or XHTML document hence these are termed as an 'embedded script'. Also, these may be contained in a separate file which is referenced by the document(s) that use it and hence termed as an 'external script'. Upon request, the essential files are sent to the user's computer by the Web server or servers on which they reside. The user's Web browser executes the script and then displays the document including any visible output from the script. Besides, client-side scripts may also include instructions for the browser to proceed in response to definite user actions, for example clicking a mouse button. These instructions can be pursued without any additional communication with the Web server or server. By viewing the file that holds the script, users can observe its source code. Due to security limits,client-side scripts may not be allowed to access the user's computer outside the Web browser application. Client-side scripting is not inherently unsafe. Users can maintain their Web browsers up-to-date to avoid exposing their computer and data to susceptible that are revealed.

The most recent group of Web browsers and Web pages have a tendency to utilize a intense quantity of client-side scripting which accounts for an enhanced user interface where the user never experiences the unfriendly 'refreshing' of the Web page.

Instead the user observes an animated GIF file indicating that the request occurred and the page will be updated in a while. AJAX or Asynchronous JavaScript And XML is an important addition to the JavaScript language which allows Web developers to communicate with the Web server in the background without needing an entirely new version of the page to be requested and rendered. Scripting languages, which can be embedded within HTML or HyperText Markup Language are generally used to add functionality to a Web page, such as different menu styles, graphic displays

or to serve dynamic and classified advertisements. These client-side scripting languages affect the data that the end user views in a browser window.

JavaScript

JavaScript is considered as the most publicized and recognized scripting language initially developed by Netscape as LiveScript which facilitates more functionality and enhancement to Web page authoring that unprocessed HTML could not hold. Later, a standard version of JavaScript was developed to work in both Netscape and Microsoft's Internet Explorer making the language to a large extent universal. This specifies that the JavaScript code can run on any platform that has a JavaScript interpreter. JavaScript, also sometimes abbreviated JS, is a prototype based scripting language that is dynamic in nature. Also, it is a multi-paradigm language supporting object oriented, imperative and functional programming styles. Officially, JavaScript was recognized in the ECMAScript language standard and is primarily used in the form of client-side JavaScript implemented as part of a Web browser to provide enhanced user interfaces and dynamic Websites. This facilitates programmatic access to computational objects within a host environment.

JavaScript is by far the most widely used language for client-side scripts. The three most significant features include that JavaScript was the first scripting language for Web browsers to gain popularity, it alone has formed the basis of a standard and it is the only language supported natively by the two browsers that share virtually the entire Web browser market—Netscape Navigator and Microsoft Internet Explorer. JavaScript offers a core object model that provides developers with a set of basic data structures and functionality. This core object model is extended in both server and client side versions. For example, client-side JavaScript extends the core object model with support for the browser and document object models.

ECMA-262 or ECMAScript



ECMA-262: A standard for scripting languages published by the European Computer Manufacturer's Association

ECMA-262 or ECMAScript is a standard for scripting languages published by the European Computer Manufacturer's Association. ECMA is a nonprofit international industry association that is engaged in the standardization of information and communications technologies. Established in 1961, ECMA has published more than 300 standards documents and numerous technical reports. Besides, ECMA standards have served as a base for many other European and international standards.

ECMA-262 is based on a mutual compliance from Netscape and Microsoft. Hence, ECMAScript incorporates elements of both JavaScript and Jscript. Netscape's first implementation of ECMA-262 was JavaScript 1.1 and Microsoft's was JScript 3.0. The implementations of JavaScript and JScript in the latest versions of Navigator/ Communicator and Internet Explorer are entirely compliant with ECMA-262. In addition, the International Standards Organization (ISO) has approved the ECMA-262 specification as ISO-16262.

Server-Side Scripting Languages

Technically, because the server-side scripting is about programming the behavior of the server so server-side scripts require that their language's interpreter be installed on the server and produce the same output regardless of the client's browser, operating system or other system details. On the contrary, server-side scripts written in languages, such as Perl, PHP, ASP.NET, Java and server-side VBScript are executed by the

Web server when the user requests a document. The output is produced in a format understandable by Web browsers, usually HTML, which is then sent to the user's computer. The user cannot perceive the script's source code unless the author publishes the code separately.

NOTES

Documents produced by server-side scripts may consecutively contain client-side scripts.

Web browsers are applications for displaying Web pages. Scripts can be run by Web browsers to change the appearance or actions of a Web page, for example to change the content to be specific to the current user. A host of special purpose languages have been specifically developed to control the operation of Web browsers which include JavaScript, VBScript by Microsoft which only works in the Internet Explorer, XUL by the Mozilla project which only works in Firefox and XSLT a presentation language that transforms XML content into a new form. Client-side scripts are sent by the server 'asis' or in its present condition and are run by the client's computer. An example of client-side scripting is a JavaScript alert box popping up when the user clicks a button on the Web page. Some languages, such as Perl were originally started as scripting languages but were later developed into programming languages suitable for wide range of purposes.

A number of languages have been specifically designed for replacing application specific scripting languages by embedding them in application programs. The application programmer includes the concept of 'hooks' so that the scripting language can control the application. JavaScript still is a language for scripting inside Web browsers however the standardization of the language as ECMAScript has made it popular as a general purpose embeddable language. Other applications embedding ECMAScript implementations include the Adobe products Adobe Flash (ActionScript) and Adobe Acrobat used for scripting PDF files. Tcl was created as an extension language but has been used more frequently as a general purpose language in roles similar to Python, Perl and Ruby. The following are some examples of server-side scripting languages with their file name extension:

- ASP (*.asp, *.aspx)
- C via CGI (*.c, *.csp)
- ColdFusion Markup Language (*.cfm)
- Java via JavaServer Pages (*.jsp)
- JavaScript using Server-side JavaScript (*.ssjs, *.js)
- Lua (*.lp *.op)
- Perl CGI (*.cgi, *.ipl, *.pl)
- PHP (*.php)
- Python via Django (*.py)
- Ruby, e.g. Ruby on Rails (*.rb, *.rbw)
- SMX (*.smx)
- Lasso (*.lasso)
- WebDNA (*.dna,*.tpl)
- Progress WebSpeed (*.r,*.w)

NOTES



PHP: PHP is a server side HTML embedded scripting language that you can use to create dynamic Web pages

Server-side scripting is thus defined as 'Web server technology in which the user's request is fulfilled by running a script directly on a Web server to generate dynamic Web pages', i.e., a server-side script runs on the server rather than your computer. When you visit a Website, the script will create the Web page on the fly (dynamically). Dynamic Websites are becoming very popular because they can be easily maintained. Websites with server-side scripts can typically be updated right on the server. The files themselves do not change, as they only contain the scripts. Popular server-side scripting languages include Perl, PHP, ASP, JSP, Ruby, ColdFusion and Python. Many server-side scripting languages are cross-platform and open for anyone to use. Specialized scripting languages include the following:

PHP: PHP is a server side HTML embedded scripting language that you can use to create dynamic Web pages. It includes predefined functions that create, open, read, write and close files stored on the server. You can create XML-based Web applications using PHP.

Some browsers, such as Netscape Communicator 4.x series do not contain XML parser. PHP overcomes this problem by supporting XML parsing. PHP provides a Document Object Model (DOM) to access XML elements, an XML extension and an eXtensible Stylesheet Language (XSL) processor to support XML parsing.

Perl or Practical Extraction and Report Language: This is a popular string processing language for writing small scripts for system administrators and Website maintainers.

Hypertalk: It is another example. It is the underlying scripting language of HyperCard.

Lingo: It is the scripting language of Macromedia Director which is an authoring system for develop high performance multimedia content and applications for CDs, DVDs and the Internet.

AppleScript: A scripting language for the Macintosh allows the user to send commands to the operating system to carry out complex data operations, for example open applications.

3.4.1 I/O Streaming

Java I /O package contains many classes, each with a variety of member data stream and methods. The Java I/O model is based on the concept of input output stream. A stream uses an abstraction that either produces or consumes information. A stream is a sequential flow of bytes from a source or to a destination. The input devices may comprise a keyboard, files, memory and the Internet, etc., and output devices may include monitor and files, etc.

The `InputStream` class is an abstract class that serves as the base class for all other input streams and defines a basic interface for reading streamed bytes of information. `The InputStream` defines three different `read ( )` methods for reading input data in various ways and their signatures are given as follows:

- `int read( )`
- `int read(byte b[ ])`
- `inte read(byte b[ ], int off, int len)`

Java input stream classes and their class hierarchy is shown as follows:

NOTES

InputStream Class

- FileInputStream class
- ByteArryInputStream class
- StringBufferInputStream class
- SequenceInputStream class
- PipedInputStream class
- FilterInputStream class
- BufferedInputStream class
- LineNumberInputStream class
- PushBackInputStream class
- DataInputStream class

Program: A program that demonstrates `InputStream` class

```
public class Example
 {
 static String ss = new String("Today upon a bus, \n"
+ " I saw a lovely girl" + " with golden hair. \n");
 static StringBufferInputStream sbis = new
StringBufferInputStream(ss);
 public static void main(String argv[ ])
    {
    int cc;
    try
      {
       while((CC = sbis.read( ))! = -1)
          {
           System.out.println((char)cc);
           }
             catch(Exception e)
        {
        System.out.println("Error:" + e.toString( ));
        }
    }
}
}
```

Output of the program:

```
 Today upon a bus,
 I saw a lovely girl with golden hair.
```

3.4.2 `OutputStream` Class

The counterpart to `InputStream` is the `OutputStream` class which provides the basic functionality for all output streams.

- `write( )` : Writes data to the stream.
- `flush( )` : Forces any buffered output to be written.
- `close( )` : Closes the stream.

The Java output stream classes and their class hierarchy is shown as follows:

`OutputStream` Class

- FileOutputStream class
- ByteArrayOutputStream class
- PipedOutputStream class
- FilterOurputStream class

- BufferedOutputStream class
- PrintStream class
- DataOutputStream class

Program 5: A program that demonstrates the `OutputStream` class

NOTES

```
import.io.*;
public class Example1
  {
  public static void main(String argv[ ])
      {
      byte buffer[ ] = new byte[64];
      try
      {
      System.in.read(buffer, 0, 64);
      }
  catch(Exception e)
           {
            System.out.println("Error:" + e.toString(
));
            }
      try
      {
            FileOutputStream   out   =   new
FileOutputStream("test.txt");
         out.write(buffer);
      }
      catch(Exception e)
          {
         System.out.println("Error:" + e.toString( ));
           }
  }
}
```

Output of the program:

When you run the program, you will get the text cursor blinking.

`you type` ***Hello World***

The program will create a file whose name is test.txt and content is

`Hello World`

3.5 APPLETS



Applet: A small Java program, is much more focused in scope than a Lfull-fledged Java applicatio

An **applet**, a small Java program, is much more focused in scope than a full-fledged Java application. What an applet does is up to you—the programmer. An applet is not run from the command line like an application. An applet can be considered as a small pre-compiled Java program that can be run using an applet viewer or a Java enabled browser.

3.5.1 Applets and Web Pages

Web pages are created using HTML files. When the Web browser running on your system visits a Web page stored on a remote system over the Internet, the corresponding HTML file is downloaded into your local system. The Web browser then deals with this HTML file. There are HTML tags that instruct the Web browser to do things. Applets are embedded in an HTML file. There is an HTML tag <applet> that will

NOTES

cause your Web browser or applet viewer to download your pre-compiled Java applet into your local system and run it.

```
<title>
A simple applet
</title>
_____
 ______
<applet code = "Example.class" width = "350" height
= "200">
</applet>
```

3.5.2 Limitations of Applets

The developers of Java have imposed some restrictions on the functionality of applets. These restrictions limit the capability of applets. However, these restrictions are required for security reasons. Applets are downloaded from the Website to your local system.

Applets are not given access to a local machine's file system. Thus, applet can neither read a file from the local machine, nor can it save a file on the local system.

In addition to these limitations, there are limitations imposed by the Web browsers. Web browsers are trusted applications. It does not damage any file or operating system on the machine it is running. Web browsers limit an applet's network connectivity. Applets are allowed network connection only with the Web server from where it is downloaded. This means that you cannot have your HTML document, applet code, data and Web browser at four different locations. Your HTML document, applet code, and the data it uses have to be in one location.

3.5.3 Applet Basics

Applet basically is a Java class defined in the java.applet package of the JDK. Applet class and its class hierarchy is as follows:

```
java.lang.Object
   java.awt.Component
      java.awt.Container
         java.awt.Panel
            java.applet.Applet
```

The Applet class, which is in the `java.awt.applet` package, inherits the properties of the Panel class, which is in `java.awt` package. The Panel class inherits the properties of the Container class in the `java.awt` package. The Container class inherits the properties of the Component class, which is again in the `java.awt` package. All these sub classing has some advantages. Each of these classes implements a variety of methods. All or part of these methods are passed down and made available to the Applet class.

As Applet is actually a class, it can be used to build other user-defined applets. This is very much evident from the structure of definition of any user-defined applet. A user-defined applet is defied as:

```
public class MyExampleApplet extends java.applet.Applet
   {
   _____
   _____
}
```

Applets inherit a great deal of functionality from the Applet class.

Check Your Progress

1. What is JavaScript?
2. What is the use of JavaScript in Web pages?
3. What is a VBScript?
4. What is the role of VBScript in HTML?
5. Define variables in VBScript.
6. What is a scripting language?
7. Define ECMAScript.
8. What is a Web browser?
9. What is PHP?

3.5.4 The Applet Class

NOTES

To create your own applet, you should create a sub class of Applet class and override some of the methods. The following are some of the methods implemented in the Applet class:

1. `public void destroy( ) / / Empty`
2. `public String getAppletInfo( )`
3. `public AudioClip getAudioClip(URL url)`
4. `public AudioClip getAudioClip(URL url, String name)`
5. `public URL getCodeBase( )`
6. `public URL getDocumentBase( )`
7. `public Image getImage(URL url)`
8. `public String getImage(URL url, String name)`
9. `public String getParameter(String name)`
10. `public String[ ] [ ]  getParameterInfio( )`
11. `public void init( )         / /Empty`
12. `public void play(URL url)`
13. `public void play(URL url, String name)`
14. `public void Start( )   / / empty`
15. `public void stop( )         / /  Empty`

The following are the methods that your applet generally overrides.

1) `getAppletInfo( )`

   This method maintains text and explains who is the owner of the applet, what versions it is, what the copyright is, and so on.

2) `getParameterInfo( )`

   It returns an arbitrary length array of 3 elements arrays of strings. Each element of this array has three elements and describes one of the parameters that this applet understands. The three elements of each parameter description are strings specifying the parameter name, type and description.

3) `init( )`

   It performs any initialization.

4) `start(  )`

   This method makes the applet to do things.

5) `stop( )`

   This stops the applet from executing.

6) `destroy( )`

   This frees up the resources that the applet is holding.

The simplest possible applet is the one that does not do anything.

```
 public class Example1 extends  java .applet.Applet
{

}
```

The HTML file for this is:

```
<html>
<applet code= "Example1.class" width="400" height="200">
</applet>
</html>
```

NOTES

When this applet is viewed through the applet viewer, you will get a blank window of dimensions 400×200 pixels on the screen.

THE HELLO WORLD APPLET

Java code for the Hello world applet:

```
import  java .awt .*;

import java.applet.*;
public class Example 2 extends java .applet.Applet
{
public void paint(Graphics g)
   {
g.draw String ("Hello World",50,50);
}
}

   The HTML file for the above applet.
<html>
<applet code="Example2.class" width="300" height="200">
</applet>
</html>
```

The Html file Example2.html for the hello world applet

DRAWING GRAPHICS

The next applet is an extension of the hello world applet. The display is made more attractive here. It demonstrates the use of colors and fonts.

Java code for the modified Hello World applet:

```
import java.awt.*;
import java .applet.*;
public class Example3 extends java.applet.applet
        {
static final String message ="Hello world";
private Font font;
public void init()
{
Font f1 =new  font("Helveltica",font.BOLD,48);
}
public void paint (Graphics g)
{
g.setColor(Color .pink);
g.fillOval(10,10,330.100);
g.setColor (Color.red);
g.drawOval(10,10,330,100);
g.drawOval(9,9,332,102);
g.drawOval(8,8,334,104);
g.drawOval(7,7,336,106);
g.setColor (Color.black);
g.setFont(font);
g.drawString(message,40,75);
```

NOTES



Socket: It is used to establish connection between a client program and a server program

The HTML file for the above applet

```
<html>
<applet code="Example3.class" width="400" height="200">
</applet>
</html>
```

3.6 SOCKET PROGRAMMING

Socket is used to establish connection between a client program and a server program. The protocols used to create sockets for communication of data are TCP/IP and UDP.

TCP/IP

Transmission Control Protocol and Internet Protocol TCP/IP consist of a set of rules for exchanging information between applications, across networks. TCP is a reliable and connection-oriented service that transmits information in the form of packets. The TCP/IP sends the packets to their intended destinations after establishing a connection with the destination and the recipient sends an acknowledgement to the sender on receiving the packets. The sender retransmits the data, if the acknowledgement is not received.

UD Protocol

User Datagram Protocol (UDP) works as a datagram service between the clients and server. Datagram is a unit of information that consists of a frame and frame in turn consists of a set of bits. UDP provides fast and connectionless transportation of packets. The UD protocol is not a reliable protocol, as it does not guarantee the delivery of packets. The data sent through the UD protocol contains the recipient's address and transfers the data without establishing a connection between the sender and the recipient. Acknowledgement is also not received in this case.

Ports and Sockets

A port is a numeric address, which forms a common channel for communication of data using a particular protocol. A 16-bit number, ranging from 0 to 65535, represents ports. Each protocol is assigned a specific port number through which a client makes a request. Various protocols, with their port numbers, are:

- Transmission Control Protocol/ Internet Protocol (TCP/IP) consists of TCP and UDP that are used to exchange the data over a network. The TCP/IP reserves the port number 1024.
- File Transfer protocol (FTP) enables the transfer of files efficiently and reliably between computers. FTP reserves the port number 21.
- Telnet provides an interactive, text-based communication between a client and server. The telnet protocol reserves the port number 23.
- Finger enables the file transfer for clients using the FTP. The finger protocol reserves the port number 79.
- Hyper Text Transfer Protocol (HTTP) enables the interaction of clients with the Internet. The HTTP reserves the port number 80.

NOTES

Ports are categorized as either well-known or ephemeral. The well-known port, also known as privileged port, is created on the server side. The port number of a well-known port is less than 1024. The ephemeral port, also known as transient or temporary port, is created on the client side when a client wants to connect to a server. The ephemeral port is chosen at random and is destroyed when the client application terminates. The port number of the ephemeral port is greater than 1023. Java uses mainly TCP and UDP protocols for making connection requests.

Figure 3.1 shows how the data is passed to an application across the network.

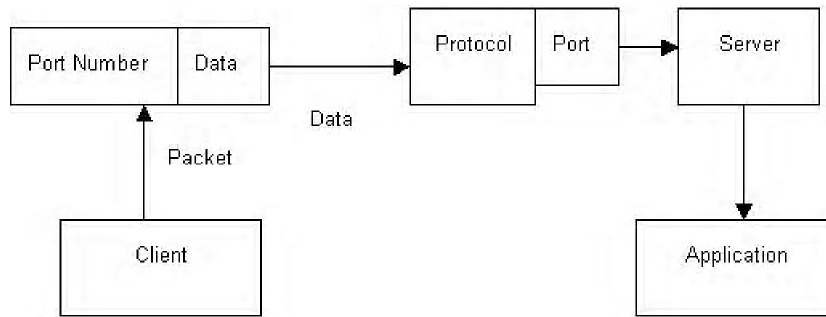


Fig. 3.1 Data Passing Over a Network

A socket establishes an uninterrupted network connection between a client and a server using the port number. The server computer has a host name and a port number that the client computer uses to make a connection request. The server accepts the connection request by creating a new socket that the server uses for communication with the client. The server has a separate local port for communication with individual clients. The client and server communicate with each other using a socket connection. Figure 3.2 shows how the connection is established between client and server.

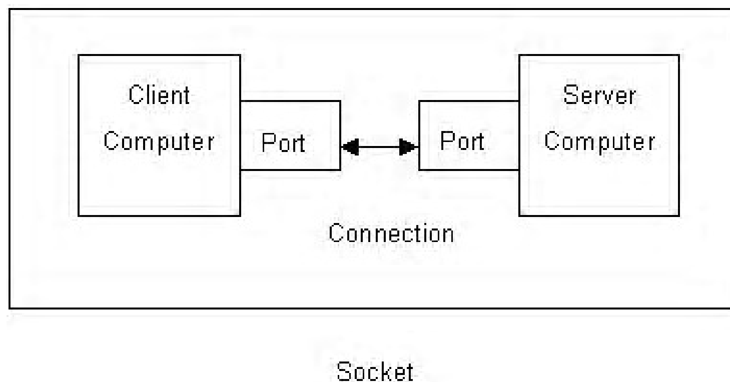


Fig. 3.2 The Connection Establishment

You create network programs in Java using the java.net package. The java.net package provides the Socket class that creates a client socket and the ServerSocket class that enables servers to listen for client requests and accept their connections. The code for implementing socket connection is:

Implementing Socket Connection;

```
Socket mySocketConnection;
try
{
mySocketConnection = new Socket("www.banner.com" , 1201);
```

```
}
catch(IOException e)
{
System.out.println("Exception is:" +e);
}
```

NOTES

The above code creates the socket object, mySocketConnection, at port number 1201. The code displays an error message, if it is unable to create a socket object.

3.6.1 Using TCP/IP Sockets

TCP/IP sockets transfer the data over the network using the TCP/IP. TCP sockets establish the connection between the host computers first and then start sending the data. This connection is terminated automatically, when the data has been sent to the specified host. Java provides two types of TCP/IP sockets, client and socket.

The client socket is used to connect the client to the server. The client socket is created when you create an object of the Socket class. The Socket class is a member of the java.net package. The Socket class provides constructors for creating the client socket. The various constructors are:

- `Socket(String HostName, int PortNumber)` creates a client socket that connects the local socket to the named host at the specified port number. This type of constructor throws two Java exceptions, UnKnownHostException and `IOException`.
- `Socket(InetAddress IPAdress, int PortNumber)` creates a client socket using the existing host, which has a specified IP address and port number. IP address is the object of the `InetAddress` class that represents the IP address of the existing host and `PortNumber` represents its port number. This type of constructor throws a Java exception `IOException`.
- `Socket()` creates an unconnected socket, with the system-default type of `SocketImpl`.
- `Socket(InetAddress host, int PortNumber, boolean stream)` deprecated. Use `DatagramSocket` instead for UDP transport.
- `Socket(InetAddress address, int PortNumber, InetAddress localAddr, int localPort)` creates a socket and connects it to the specified remote address on the specified remote port.
- `Protected Socket(SocketImpl impl)` creates an unconnected Socket with a user-specified `SocketImpl`.
- `Socket(String host, int PortNumber, boolean stream)` deprecated. Use `DatagramSocket` instead for UDP transport.
- `Socket(String host, int PortNumber, InetAddress localAddr, int localPort)` creates a socket and connects it to the specified remote host on the specified remote port.

You can examine the status and the input/output stream associated with the socket using the socket methods.

Table 3.2 describes the various socket methods.

NOTES

Table 3.2 Socket Methods

Method	Description
`getInetAddress()`	Returns the IP address of the host computer.
`getPort()`	Returns the port number of the host computer.
`getLocalPort()`	Returns the port number of the local host.
`getInputStream()`	Returns the input stream that is associated with the socket.
`getOutputStream()`	Returns the output stream that is associated with the socket.
`Close()`	Closes the socket connection.

The server socket is used for creating a server application, which enables a server to communicate with the client. The server socket is created using the ServerSocket class. The ServerSocket class is a member of the java.net package.

- `ServerSocket(int PortNumber)` creates a server socket for the specified port number that has a fixed queue length of 50. The queue length determines how many client connections the system can accept at a time.
- `ServerSocket(int PortNumber, int MAXQueue)` creates a server socket for the specified port number with the specified queue length. The MAXQueue parameter represents the maximum queue length as integer type.
- `ServerSocket(int PortNumber, int MAXQueue, InetAddress LocalAddress)` creates a server socket for the specified port number and the specified local host address. LocalAddress represents the object of the InetAddress class that specifies the IP address of that local host.

You can examine the status and the input/output stream associated with the server socket using the server socket methods.

Table 3.3 describes the various server socket methods.

Table 3.3 Server Socket Methods

Method	Description
`Socket accept()`	Finds a connection to be made to this socket and accepts it.
`void bind(SocketAddress endpoint)`	Binds the ServerSocket to a specific address (IP address and port number).
`int getLocalPort()`	Returns the port on which this socket is listening
`boolean isClosed()`	Returns the closed state of the ServerSocket.
`void bind(SocketAddress endpoint, int backlog)`	Binds the ServerSocket to a specific address (IP address and port number).
`ServerSocketChannel getChannel()`	Returns the unique ServerSocketChannel object associated with this socket, if any.
`InetAddress getInetAddress()`	Returns the local address of this server socket.
`SocketAddress getLocalSocketAddress()`	Returns the address of the endpoint this socket is bound to, or null if it is not bound yet.
`int getReceiveBufferSize()`	Gets the value of the SO_RCVBUF option for this ServerSocket, that is the proposed buffer size that will be used for Sockets accepted from this ServerSocket
`boolean isBound()`	Returns the binding state of the ServerSocket.

NOTES

Creating a Server Program

The server program creates a server socket that can communicate with the client socket using TCP/IP. The TCPServer class creates a server socket by using an object of the `ServerSocket` class. The following program code shows how to use the `ServerSocket` class to create a server socket:

Creating a Server Socket

```
import java.net.*;
import java.lang.*;
import java.io.*;
public class TCPServer
{
    public static final int PORT = 1050;
    public static void main( String args[])
    {
       ServerSocket server_socket = null;
    Socket socket = null;
    System.out.println(" Server is waiting.... ");
    try
    {
           server_socket =  new ServerSocket(PORT);
           System.out.println("TCPServer has been started
:"+ server_socket);
            try
            {
               // Creates a socket connection using TCP/IP
               socket = server_socket.accept();
                  System.out.println("TCPClient has been
connected :"+ socket);
               // Receives the message from the TCPclient
system
                        DataInputStream  dstream  =  new
DataInputStream(socket.getInputStream());
              // Sends the message to the TCPclient system
                 PrintStream pstream = new   PrintStream
(socket.getOutputStream());
              pstream.println("Hello! TCPServer here.");
               pstream.close();
              socket.close(); // Terminate the connection
             }
            catch(SocketException se)
             {
               System.out.println("Error in server socket
"+ se.getMessage());
             }
            catch(Exception e)
             {
                    System.out.println("Error  in  server
starting"+ e.getMessage()) ;
             }
```

```
  System.out.println(" Connection from : " +
socket.getInetAddress());
         }
        catch(Exception e)

         {
               System.out.println("Error in TCPServer "+
e.getMessage()) ;
         }
     }   // End main()method
}  // End of TCPServer class
```

NOTES

The above code shows how to create a server socket at port number `1050`. The `ServerSocket(PORT)` creates a server socket of specified port number. The server socket can communicate with the socket using the methods, `getInputStream()` and `getOutputStream()`.

Creating a Client Program

The client program creates a client socket that can communicate with the server socket using TCP/IP. The `TCPClient` class creates a client socket using an object of the Socket class. The following program code shows how to use the Socket class to create the client socket:

```
import java.lang.*;
import java.io.*;
import java.net.*;
import java.net.InetAddress;
class TCPClient
{
    public static void main(String args[])
    {
       Socket socket = null;
        DataInputStream dstream = null;
        PrintStream pstream = null;
           System.out.println(" Client is connecting to
TCPServer....");
        try
        {
                                InetAddress ipadss
=InetAddress.getByName("java.sun.com");
         socket = new Socket(ipadss,1050);
                                     pstream = new
PrintStream(socket.getOutputStream());
           pstream.println(" Hi! TCPClient here.");
                     DataInputStream istream = new
DataInputStream(socket.getInputStream());
           System.out.println(istream.readLine());
        }
       catch(SocketException se)
```

```
            {
               System.out.println("Error in sockets " + se);
            }
          catch(IOException ioe)

  {
             System.out.println("Input/Output Errors " + ioe);
            }
           finally
          {
             try

             {
                socket.close(); // Closes the socket
             }
             catch(IOException ioe)
             {
                System.out.println(" Socket closing Errors:" +
  ioe.getMessage());
             }
          } // End finally block
       } // End the main() method
  } // End the TCPClient class
```

NOTES

The above code shows how to create a client socket that can communicate with the server at port number 1050. The client socket communicates with the server using the methods, `getInputStream()` and `getOutputstream()`. You can run the client and server program after successfully compiling both the programs. Figure 10.3 shows the output of above program.

```
C:\TCP>java TCPServer
 Server is waiting....
TCPServer has been started :ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=1
050]
TCPClient has been connected :Socket[addr=192.168.0.205/192.168.0.205,port=1407,
localport=1050]
 Connection from : 192.168.0.205/192.168.0.205
```

Fig. 3.3 *Running TCPServer*

Figure 3.4 shows the output of the client and server program.

```
C:\TCP>java TCPClient
 Client is connecting to TCPServer....
Hello! TCPServer here.
```

Fig. 3.4 *Running TCPClient*

3.6.2 Using UDP Sockets

UDP sockets exchange data over the network by using the User Datagram Protocol (UDP). UDP is a connectionless protocol that can send data in the form of data bundles called as datagram that consists of the destination address. UDP is a less reliable protocol than TCP/IP because it does not create a connection between the host computer and client. UDP sockets are used for sending data without establishing a connection between the host computers. Java provides two classes for creating a UDP socket, `DatagramPacket` and `DatagramSocket`.

The DatagramPacket class creates the datagram packet that stores information in the form of an array. The datagram packets can be transmitted over the network using the UDP. Java provides constructors for the DatagramPacket class. The various constructors are:

NOTES

- DatagramPacket(byte []buf, int length) creates the datagram packets of specified length. The parameter, data[] represents an array that stores the information. The parameter, length, indicates the maximum size of packet that can be transmitted over the network.
- DatagramPacket(byte data[]buf, int length, InetAddress IPAddress, int port) creates the datagram packets of specified length, IP address and port number. The parameter, IPAddress, represents the IP address of the target host and parameter port represents the port number of the target host.
- DatagramPacket(byte[] buf, int offset, int length) creates a DatagramPacket for receiving packets of length length and specifies an offset into the buffer.
- DatagramPacket(byte[] buf, int offset, int length, InetAddress address, int port) creates a datagram packet for sending packets of length length with offset ioffsetto the specified port number on the specified host.
- DatagramPacket(byte[] buf, int offset, int length, SocketAddress address) creates a datagram packet for sending packets of length length with offset ioffsetto the specified port number on the specified host.
- DatagramPacket(byte[] buf, int length, SocketAddress address) creates a datagram packet for sending packets of length length to the specified port number on the specified host.

You can send or receive datagram packets across the network using the datagram packet methods. There are several methods for accessing the state of the datagram packet. Table 3.4 describes the various datagram packet methods.

Table 3.4 *DatagramPacket Methods*

Method	Description
send(DatagramPacket dp)	Sends the datagram packet to the specified host.
Receive(DatagramPacket dp)	Retrieves the datagram packet from the specified host.
GetLength()	Returns the length of the datagram packet.
getAddress()	Returns the IP address of local host.
getPort()	Returns the port number of the specified host.

The DatagramSocket class creates a socket that controls the transmission of datagram packets across the network. The object of the DatagramSocket class throws an exception, SocketException if any exception occurs at the time of datagram packet transmission over the network. Java provides DatagramSocket constructors for creating the socket. The various constructors are:

- DatagramSocket() creates a default socket that transmits a datagram packet across the network.

NOTES

- `DatagramSocket(int port)` creates a socket that can transmit a datagram packet to the host of specified port number across the network.
- `Protected DatagramSocket(DatagramSocketImpl impl)` creates an unbound datagram socket with the specified `DatagramSocketImpl`.
- `DatagramSocket(int port, InetAddress laddr)` creates a datagram socket, bound to the specified local address.
- `DatagramSocket(SocketAddress bindaddr)` creates a datagram socket, bound to the specified local socket address.

You can send or receive datagram sockets across the network using the datagram socket methods. There are several methods for accessing the state of the datagram sockets. Table 3.5 describes the various datagram socket methods.

Table 3.5 *`DatagramSocket` Methods*

Method	Description
`bind(SocketAddress addr)`	Binds this DatagramSocket to a specific address & port.
`connect(InetAddress address, int port)`	Connects the socket to a remote address for this socket
`InetAddress getLocalAddress()`	Gets the local address to which the socket is bound.
`getInetAddress()`	Returns the address to which this socket is connected.
`getPort()`	Returns the port number of the socket
`close()`	Closes the socket connection.
disconnect()	Disconnects the socket.
send(DatagramPacket p)	Sends a datagram packet from this socket.

3.7 LOOKING UP INTERNET ADDRESS

Using Internet has become common nowadays. We will now understand how the Internet interprets the Internet address. For instance, the Internet addresses are written as www.hotmail.com. Let us write one more address as server.institution.domain. It is significant to note that the address www.hotmail.com is not the actual address. It is actually a text version of the Internet address, which is basically a binary representation. Now we compare www.hotmail.com and server.institution.domain. Here, www is the name of the server owned by the institution (in this case, it is hotmail) and this server is connected to the Internet and a domain server (com in this case) which maintains a database of the addresses of different servers using the same domain com. The domain name has no geographical relevance, and two sites with the same domain name may exist at two different ends of this world.

The above mentioned case is the simplest one. Now, let us take the case of a large organization, which may have several other servers for different purposes, such as Web server, e-mail server, print server, etc. Now, take this example of www.sun.planet.universe.in. This address is written in five parts separated by three dots. This address apparently indicates that a group Planets (planet) comes under an Universe sub domain which is a part of India domain and maintaining one server sun out of many servers, which is linked to Internet through its Web server. Similarly, an organization with several departments may create addresses for its sub-domain with different servers being maintained there.

The Internet is the collection of several independent networks, which are interconnected with one another. Each independent network may have several hosts. Keeping this in mind, you can now think of your address. Your house has a unique house number, which is not assigned to any other house in your locality. In this case, your house can be considered as the host, your as the network and your city as the domain. You can write your address in Internet addressing notation as houseno.locality.city. In case you want to acquaint a foreigner with your address, then you need to add your country name in your address. In this case it will become houseno.locality.city.country. Now if someone wants to send you a letter or visit your house, then is required to come to your country first and then to your city. Only after fulfilling these two obligations, he can reach your locality and then your house by locating your house number. The same analogy applies in the case of Internet addressing.

A host on the Internet has two significant functions, namely the identification of the network and identification of the host on the network. Therefore, the address of the host comprises of two parts namely the network address and the host address. These two parts together make 32-bit long IP address for a particular host on the Internet. The IP address is written in four octets each separated by a dot. It may have a form like 197.23.207.10. Presently, we are using IP address version 4 (IPv4). However, IP address version 6 (IPv6) has not yet been implemented.

NOTES

3.7.1 IPv4 Addressing

IPv4 addresses are uniquely used as identifiers, which work at network layer to identify the source or destination of IP packets. At present, the version of IP, which is in use, is called IPv4. In this version, every node on the Internet may have one or more interfaces and we are required to identify each of these devices with a unique address assigned to each of them. It means that each node is assigned one or more IP addresses to invoke TCP/IP. These are logical addresses and have 32 bits.



IPv4 addresses: Uniquely used as identifiers, which work at network layer to identify the source or destination of IP packets

Technically, IP addresses are expressed using binary notation with 32-bit long string. In order to recall these strings easily, dotted decimal notations are used, in which periods or dots separate four decimal numbers from 0 to 255 representing 32 bits. As there are 32 bits, each decimal number contains 8 bits and is referred to as octet.

For instance, the IPv4 address 11000000101010000000101000011001 is expressed as 192.168.10.25 in dotted decimal notation. Given below are the steps to convert an IPv4 address from binary notation to dotted decimal notation:

- Break 32-bit long address into segments of 8-bit blocks: 11000000 10101000 00001010 00011001
- Write the decimal equivalent to each segment: 192 168 10 25
- Separate the blocks with periods: 192.168.10.25

Following representation shows the structure of IP address.

11000000	10101000	00001010	00011001
192	168	10	25

Dotted Decimal Notation

We have seen that IPv4 address is expressed as a 32-bit number in dotted decimal notation. IP addresses may have fixed part and variable part depending upon the

NOTES



Multicast: It is used for one or more network interfaces located on various subnet

allocation of total addresses to an organization. The fixed part of the address may range from one octet to three octets and the remaining octets will then be available for the variable part. An IPv4 address is assigned using these parts. All bits in the fixed octet (s) are set to 1, whereas the variable octet(s) are set to 0 bits. Thereafter, convert the result into dotted decimal notation. For instance, you may take an IP address as 192.168.10.25. Now, set all fixed bits to 1 and set all variable bits to 0. This gives 11111111 11111111 00000000 00000000. After converting it into dotted decimal notation, the result is 255.255.0.0. This dotted decimal notation with fixed and variable parts is used as an address prefix to 192.168.10.25 and is expressed as 192.168.10.25/ 255.255.0.0. This way of expressing the prefix length as a dotted decimal number is known as network mask or subnet mask notation.

3.7.2 Classification of IPv4 Addresses

Internet standards allow the following addresses:

1. **Unicast —** It is assigned to a single network interface located on a specific subnet and it facilitates one-to-one communication. This is unique address is used globally for the identification of a device on the network. It may be understood as the house number in a particular locality. It includes a subnet prefix and a host ID portion.
    - **Subnet Prefix —** The subnet prefix is basically a network identifier or a network address portion of an IP unicast address. It should be noted that all nodes on the same physical or logical subnet must use the same subnet prefix, which eventually becomes unique within the entire TCP/IP network.
    - **Host ID —** The host ID, which is the host address portion of an IP unicast address, serves to identify a network node to which some devices are interfaced. It is also regarded as unique within the network segment.
2. **Multicast —** It is used for one or more network interfaces located on various subnets. It facilitates one-to-many communication. It delivers single packets from one source to many destinations. These addresses are a part of Class D addressing scheme.
3. **Broadcast —** It is allocated to all network interfaces located on a subnet and is used for one-to-everyone communication on a subnet. It delivers packets from one source to all interfaces on the subnet. Broadcast addresses may be further classified as network broadcast, subnet broadcast, all subnets directed broadcast and limited broadcast.

Internet Addresses are further classified into different categories. The number bits are used for the address prefix of a single subnet and the number of bits are used for the host ID. Therefore, it allocates the number of networks and the number of hosts per network. There are five address classes which are given below: -

- **Class A —** It uses an 8-bit network number whose first bit is always zero as shown in Table 2.6. It is reserved for IP unicast addresses. This class is used in case there are myriad hosts on a network. It uses only one octet to define the prefix length. The number of networks, which can be accommodated, are 2^8 or 128. However, out of these 128 addresses, 2 are used for administrative purposes and 126 addresses are available as prefix length. The remaining 3 octets are used for identifying up to 2^{24} or 16,777,214 host IDs.
- **Class B —** It uses 16 bits for both the network address and the host address. In this case, the first two bits are always 10. It is reserved for IP unicast addresses.

NOTES

It uses 2 octets for a particular network while the remaining two octets are deployed by the host IDs. They are particularly used for medium to large-sized networks. The Class B addresses can cater to 16,384 networks with up to 65,534 hosts per network.

- **Class C —** It is reserved for IP unicast addresses. It is meant for small networks. The first 3 octets specify a particular network and the last octet specify the host IDs**.** The Class C addresses may be used by up to 2,097,152 networks with up to 254 hosts per network. Its first three bits are always set to 110.
- **Class D —** Class D signifies IP multicast addresses.
- **Class E —** These addresses are reserved for experimental purposes.

Table 3.6 represents IPv4 addresses classifications.

Table 3.6 *Classifications of IPv4 Addresses*

32 bit address					Number of possible networks	Maximum number of host or nodes
Classification	Octet 1	Octet 2	Octet 3	Octet 4		
Class A	0bbbbbbb	xxxxxxxx	xxxxxxxx	xxxxxxxx	2^7 = 128	2^{24} = 6,777,216
Class B	10bbbbbb	bbbbbbbb	xxxxxxxx	xxxxxxxx	2^{14} =16,384	2^{16} = 65,536
Class C	110bbbbb	bbbbbbbb	bbbbbbbb	xxxxxxxx	2^{21}=2,097,152	2^8 = 256
Class D	1110bbbb followed by a 28 bit multicast address					
Class E	1111; reserved					

3.8 E-MAIL CLIENT

Electronic mail or e-mail is one of the most popular applications of the Internet today. It refers to the process of transmission of messages in a communication network which was started by Ray Tomlinson in 1971.

In Web-based e-mail services, the mail resides in the server and is accessed using the Web browser. The user can access his e-mails after he has logged onto the server Web site. Messages can be sent and received from a single e-mail address to another e-mail address or can be broadcasted, that is, the same message can be sent to many recipients.

The exchange of messages between two terminals or hosts is done by using the SMTP along with software programs known as Mail Transfer Agents (MTAs). These e-mails can be stored either on one of the terminals— the client terminal, the server terminal or at both the places. Post Office Protocol (POP) and Internet Message Access Protocol (IMAP) are standard protocols that are used to retrieve messages from the server. The standard mail box formats available are maildir and mbox. E-mail clients can also use their own proprietary mailbox formats. However, while using these they must ensure that they have the necessary conversion software.

The various components of an e-mail are as follows:

- User agent.
- Message transfer agent.
- Message access agent.

3.8.1 User Agent

The User Agent (UA) provides the services which are shown in Figure 3.5.

NOTES

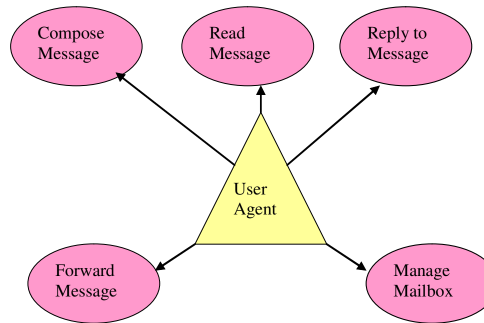


Fig. 3.5 *User Agent Services*

The various services provided by the UA are as follows:

1. **Composing Messages:** UA provides built-in text editors or templates to compose, edit, spell check, vocabulary check and perform other sophisticated word processing tasks.
2. **Reading Message:** When the messages are received, the UA reads them. Whenever there is a new e-mail message, the UA shows a summary or preview of the received message (before the message is actually opened by the user). It also gives information about the serial number of the message and status of the e-mail, such as new, read or unread, the message size, sender's name, subject of e-mail, etc.
3. **Message Reply:** A user agent is used to reply to the message of the sender of the e-mail.
4. **Message Forwarding:** A message can either be sent to a third party as it is or with some updates.
5. **Manages Mailbox:** The UA maintains two mailboxes, inbox (all the received e-mails unless they are deleted by the user) and outbox (all the sent e-mails until the user deletes them). In addition, it is possible to create some customized mailboxes.



Message reply: A user agent is used to reply to the message of the sender of the e-mail

3.8.2 Message Transfer Agent

Message Transfer Agents (MTA) are programs that actually transfer mails. A system generally needs two MTAs—the client MTA to send mails and a server MTA to receive mails. SMTP defines these two MTAs over the Internet. On the UNIX platform, for setting the SMTP server, the command 'sendmail' is used and on Windows platforms, Windows Exchange is used.

When someone sends an e-mail, the SMTP is referred to for the first time between the sender and the mail server of the sender. It is referred to the second time when it is between the mail server of the sender and that of the receiver, as shown in Figure 3.6. This implies that in an e-mail transfer, the SMTP is referred to twice.

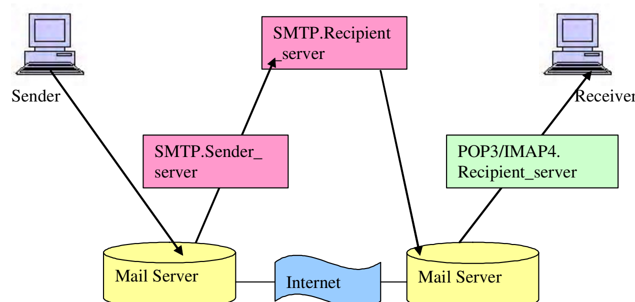


Fig. 3.6 *E-Mail Transfer from Sender to Receiver*

3.8.3 E-Mail Message Format

Internet e-mail messages consist of the following two major sections:

- **Message Header:** This part contains fields, such as subject, sender's name, receiver's name and other information about the e-mail.
- **Message Body:** This is the actual message content in text form. This can also have a signature at the end of the e-mail, quiet like the one in snail mails. A blank line separates the header from the body.

Message Header

Every e-mail message has a header along with several fields. Each of these fields has a name and a value. Every field name starts with a character of the line and ends before the separator character ':'. The field value always follows the separator and can be considered as the 'body' of the field. All the field names and corresponding values can have only 7-bit ASCII characters. Multipurpose Internet Mail Extension (MIME) encoded words can be used to represent non-ASCII values.

The message header of an e-mail contains the following fields:

- **From:** The sender's e-mail address. It may also contain the names of the message senders.
- **To:** The receiver's e-mail address; it may also contain names of the recipient(s).
- **Subject:** A brief, summarized topic of the message.
- **Date:** The time and date when the message was written. This field is automatically filled in while sending. The recipient's client may display the time and date in the format which is as per his local browser.
- **Message-ID:** This is also an automatically generated field and can be used to prevent multiple delivery and for reference in In-Reply-To: cases.

Other common header fields for e-mail include the following:

- **Bcc** (Blind carbon copy): These are addresses added to the delivery list but are not listed in the message data. It remains invisible to other recipients.
- **Cc** (Carbon copy): These are addresses added to the delivery list and also listed in the message data. All the recipients addresses are visible to each other
- **Content-Type:** It provides information about how the message is to be encoded and displayed. It is usually a MIME type.
- **In-Reply-To:** It is the message-ID of the message to which the e-mail is a reply and is used to link related messages together.
- **Received:** This tracks information generated by mail servers that have previously handled a message, in reverse order, that is, from last receiver to first.
- **Reply-To:** It is the e-mail address that is used to reply to the e-mail.
- **References: It** includes the message-ID of the message to which the e-mail is a Reply-to.
- **Sender:** This is the address of the sender or someone on behalf of the author listed in the **From** field (manager, assistant, etc.).

Message Body

Nowadays, either plain text or HTML can be used to create e-mails having graphical features enabled in the message body. A self-generated plain text copy of the HTML e-mail message is automatically included in the e-mails for further editing.

NOTES

Using HTML presents several advantages. Some of them are as follows:

- Inline links and images in messages can be incorporated.
- Previous messages can be differentiated by setting them in block quotes
- Signatures can be added to messages.
- Features like different font styles, Underline and italics can be extensively used.

The disadvantages of HTML include increased size of e-mail, virus attacks etc. HTML e-mail is prone to attacks from malicious software or 'malware'.

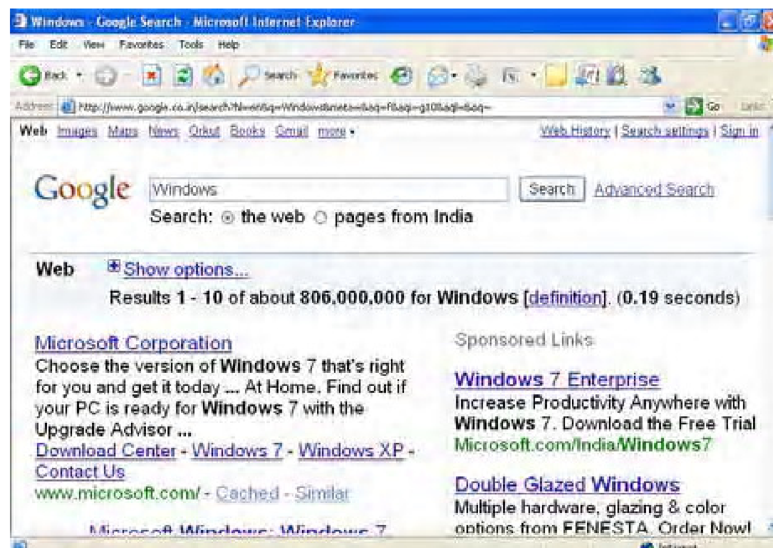
3.9 WEB PAGE RETRIEVAL

The Internet is considered as hub of information as it provides vast amount of data, such as text, audio, video and other documents related to any topic or subject. Users need to be able to find relevant information to satisfy their particular information needs. There are two ways of searching for information: to use a search engines or to browse directories organized by categories, such as Yahoo Directories. A large part of the Internet is not accessible, for example private databases and intranets.

Information Retrieval (IR) or the Web page retrieval is the task of representing, storing, organizing and offering access to information. The first IR system implemented in 1970's was designed to work with small collections of text, for example legal documents. Some of these techniques are now used in search engines.

Search Engine

Search engine improves the user's browsing experience. All those users who use Google search engine can get the quick and fast information. If user hits the **'I am feeling lucky button'** on the popular search engine, the button takes the user directly to the number are where user put the query fetched. But sometimes, it does not happen and ironically it is true for sending the result or Web. Speedy search engines might not be precise enough. Ambiguity creeps in because of getting the result as per query asked or the information by the users. The role of keyword is very important at this stage. For example, if you type 'Windows' keyword in the search bar and click 'search' button, you will get a list of available results that ranges from the Microsoft software to the home Windows.



Most of the search engines are aware of this approach and Google recently started the pre-empting task followed by user that suggests the combination of keyword for fetching the best result. The basic fundamental concept works with search engine is that how best to divine the user's intent and provide the exact result what user is searching for. Google search is based on the mechanism of algorithm. No matter, it is complex mathematical formula that crawls via unlimited Web sites. These sites are on the net and shown by the ranks according to pages (relevant and well informed information). Thereby, index is created to top and deliver the search results. Google search engine blends listings from the news, well-informed documents, and images, local book search engines that deliver the exact and comprehensive results. Bing, the Microsoft's search portal, was launched in year 2009. It works as the 'decision engine'.

Popular search engines contend with the rapid increasing verticalization of the Web especially in the field of tour-travel focuses and delvers the best possible results. The effects are visible in Search Engine Optimization or SEO. In this, the mechanism is used with the help of Web masters to ensure the Web sites get higher ranking in the searching the information. Sometimes, Web masters use spam their code with keywords that impacts on getting recognition from the algorithm of social engines is becoming smarter and complex so that it can integrate with the artificial intelligence. The searching domain is getting more user oriented too and will be continuing as it is being believed by Microsoft. A Web search engine is designed to search for information on the WWW and FTP servers. The search results are generally presented in a list of results often referred to as Search Engine Results Pages or SERPS. The information may consist of Web pages, images, information and other types of files. Search engines are the Softwares that enable searching of the content available on the Internet. A search engine is an information retrieval system which is used to access and retrieve information stored in WWW or a computer system attached to the Internet. Search engines also help minimize the time required to find the relevant information on the computer system. The computer system could be a standalone system or it could also be attached to the Internet. The search engines are popular amongst people as Web search engines help explores information on the World Wide Web.

Search engines are the interface to a group of contents, which allow the users to type in the keywords, so that the engine can find several matching contents to the corresponding keywords out of millions of Web pages. The keywords provided by the user are referred to as a search query. Several styles of search query syntax are used by the net users. Search query differs for different types of search engines, that is, some search engines enable users to enter two or three words separated by space, whereas others may require users to provide entire documents, pictures, sounds, and various forms of languages. Some search engines attempt to enhance the search queries to provide a quality set of items through a process known as query expansion.

http://en.wikipedia.org/wiki/Image:Search-engine-diagramen.svg

Protocol handler is a feature of the Windows Search engine that is used to communicate with and enumerate the contents of stores, such as the file system, Messaging Application Program Interface (MAPI) e-mail database, and the CSC or offline files database. Like IFilters, protocol handlers are also extensible. The protocol handler mechanism is an API that enables programmers to add arbitrary URL schemas to the existing set of command URLs by writing additional protocol handlers for them. A content handler defines what happens with a content part in response to specific events, such as when the part is activated. The content handler enables you to perform actions at particular moments in the lifecycle of the content item. It also enables you to set up data repositories and manipulate the data model prior to rendering the content item.

Check Your Progress

10. How are Web pages created?
11. State a limitation of applet.
12. Define TCP/IP.
13. Why do we use a server socket?

3.10 REMOTE METHOD INVOCATION

NOTES



Remote Method Invocation (RMI): It is a part of a Java Development kit or JDK, and allows us to develop distributed application

Remote Method Invocation (RMI) is a part of Java Development Kit (JDK). It allows us to develop distributed application. Distributed systems require computations that are running in different address spaces, particularly on different machines, to be able to communicate between one another. Java RMI facilitates such a communication specifically for Java applications. RMI is platform independent because Java is platform independent. RMI can communicate from one JVM to another. In RMI, the application is divided into small objects. One object communicates with another object through an interface. An interface is used to access the remote object and its methods. To develop the distributed application using RM, programmer is bound to follow some steps:

- Define the interfaces.
- Implementing these interfaces.
- Compile the interfaces and their implementation with Java compiler.
- Compile the server implementation with RMI compiler.
- Run the RMI registry.
- Run the server application.
- Run the client application.

RMI applications are divided into two separate programs: a server side application and a client side application. Server application creates a number of remote objects whereas the client application looks up the remote object by its name in the server's registry and then invokes a method of the server object. Each remote object of the Figure 3.7 server is identified by its remote interface.

Architecture of RMI

The complete RMI system is organized as a four layer model. Each layer can perform specific functions like establish the connection, marshal and unmarshal the parameters and transmitting the objects. The Layers are:

- Layer-1 (Application Layer)
- Layer-2 (Proxy Layer)
- Layer-3 (Remote Reference Layer)
- Layer-4 (Transport Layer)

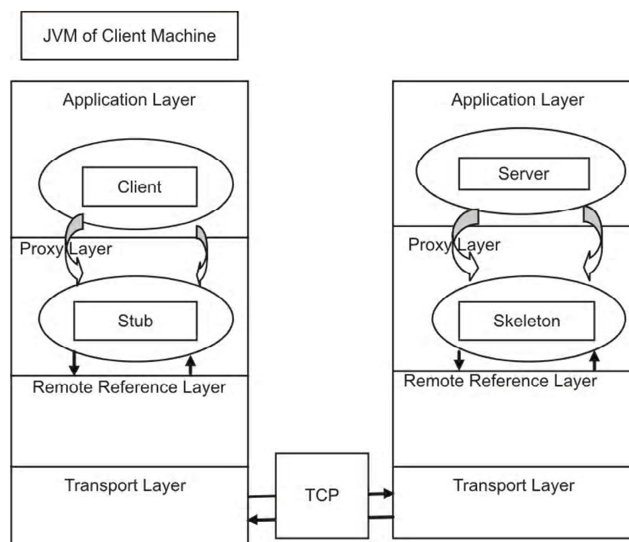


***Fig 3.7** Architecture of RMI*

All distributed application use the registry to keep track of the name of the remote objects. RMI communicates between one JVM to another JVM. When the client wants to access the remote object that resides in the server, it cannot communicate directly with the server.

3.10.1 Java Architecture RMI

The design goal for the RMI architecture was to create a Java distributed object model that integrates naturally into the Java programming language and the local object model. RMI architects have succeeded in creating a system that extends the safety and robustness of the Java architecture to the distributed computing world.

Interfaces: The Core of RMI

The RMI architecture is based on one important principle which states that the definition of behaviour and the implementation of that behaviour are separate concepts. RMI allows the code that defines the behaviour and the code that implements the behaviour to remain separate and to run on separate JVMs.

This fits nicely with the needs of a distributed system where clients are concerned about the definition of a service and servers are focused on providing the service.

Specifically, in RMI, the definition of a remote service is coded using a Java interface (refer Figure 3.8). The implementation of the remote service is coded in a class. Therefore, the key to understanding RMI is to remember that interfaces define behaviour and classes define implementation.

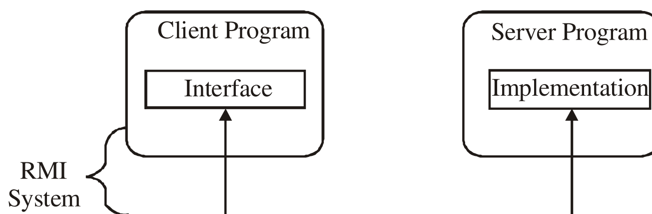


Fig. 3.8 *Overview of Getting the* `ResultSet`

3.10.2 RMI Architecture Layers

The RMI implementation is essentially built from three abstraction layers. With an understanding of the high level RMI architecture, take a look to see its implementation (refer Figure 3.9).

The first is the Stub and Skeleton layer, which lies just beneath the view of the developer. This layer intercepts method calls made by the client to the interface reference variable and redirects these calls to a remote RMI service.

The next layer is the Remote Reference Layer. This layer understands how to interpret and manage references made from clients to the remote service objects. In JDK 1.1, this layer connects clients to remote service objects that are running and exported on a server. The connection is a one-to-one (unicast) link. In the Java 2 SDK, this layer was enhanced to support the activation of dormant remote service objects via Remote Object Activation.

The transport layer is based on the Internet Protocol suite, commonly known as TCP/IP connections between machines in a network. It provides basic connectivity as well as some firewall penetration strategies.

NOTES

Check Your Progress

19. Fill in the blanks with appropriate words.
   (a) __________ can also be used to create applications that run directly on a workstation running Microsoft Windows.
   (b) VBScript is inserted into HTML, between standard ______ and </script> tags.
   (c) A _______ procedure is a series of statements, enclosed by the Function and End Function statements.
   (d) Web browsers are applications for displaying ________.
20. State whether the following statements are true or false.
   (a) PHP is a server side HTML embedded scripting language that you can use to create dynamic Websites.
   (b) A scripting language for the Macintosh allows the user to send commands to the operating system to carry out complex data operations, for example open applications.
   (c) The protocol handler mechanism is an API that enables programmers to add arbitrary URL schemas to the existing set of command URLs by writing additional protocol handlers for them.
   (d) The protocol handler mechanism is an API that disables programmers to add arbitrary URL schemas to the existing set of command URLs by writing additional protocol handlers for them.

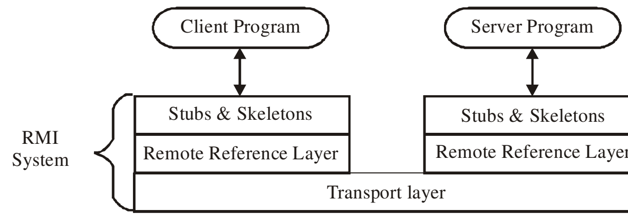


Fig. 3.9 *RMI Architecture Layers*

By using a layered architecture, each of the layers could be enhanced or replaced without affecting the rest of the system. For example, the transport layer could be replaced by a UDP/IP layer without affecting the upper layers.

Stub and Skeleton Layer

The stub and skeleton layer of RMI lie just beneath the view of the Java developer. In this layer, the Proxy pattern, an object in one context is represented by another (the proxy) in a separate context. The proxy knows how to forward method calls between the participating objects.

A skeleton is a helper class that is generated for RMI to use. The skeleton understands how to communicate with the stub across the RMI link. The skeleton carries on a conversation with the stub as it reads the parameters for the method call from the link, makes the call to the remote service implementation object, accepts the return value and then writes the return value back to the stub.

Remote Reference Layer

The Remote Reference Layers defines and supports the invocation semantics of the RMI connection. This layer provides a `RemoteRef` object that represents the link to the remote service implementation object.

The stub objects use the `invoke()` method in `RemoteRef` to forward the method call. The `RemoteRef` object understands the invocation semantics for remote services.

Other types of connection semantics are possible, for example, with multicast, a single proxy could send a method request to multiple implementations simultaneously and accept the first reply (this improves response time and possibly improves availability). In the future, Sun Microsystems may add additional invocation semantics to RMI.

Transport Layer

The Transport Layer makes the connection between JVMs. All connections are stream-based network connections that use TCP/IP. Even if two JVMs are running on the same physical computer, they connect through their host computer's TCP/IP network protocol stack.

The RMI transport layer is designed to make a connection between clients and server, even in the face of networking obstacles.

While the transport layer prefers to use multiple TCP/IP connections, some network configurations only allow a single TCP/IP connection between a client and server (some browsers restrict applets to a single network connection back to their hosting server).

In this case, the transport layer multiplexes multiple virtual connections within a single TCP/IP connection.

3.11 SUMMARY

NOTES

- JavaScript is an object oriented scripting language used to enable programmatic access to objects within both the client application and other applications.
- JavaScript is the most popular scripting language on the Internet, and works in all major browsers, such as Internet Explorer, Firefox, Chrome, Opera, and Safari.
- VBScript is acronym of Visual Basic Scripting Edition and is referred as an Active Scripting language specifically developed by Microsoft modeled on Visual Basic.
- VBScript can also be used to create applications that run directly on a workstation running Microsoft Windows.
- VBScript is a scripting language or more specifically a scripting environment that can enhance HTML Web pages and makes them more active than a simple static display.
- Scripts can be written and executed without explicit compile and link steps since they are typically created or modified by the user executing them.
- Client-side scripts are embedded within an HTML or XHTML document hence these are termed as an 'embedded script'.
- The most recent group of Web browsers and Web pages have a tendency to utilize a intense quantity of client-side scripting which accounts for an enhanced user interface where the user never experiences the unfriendly 'refreshing' of the Web page.
- Documents produced by server-side scripts may consecutively contain client-side scripts.
- Web browsers are applications for displaying Web pages. Scripts can be run by Web browsers to change the appearance or actions of a Web page, for example to change the content to be specific to the current user.
- Server-side scripting is thus, defined as 'Web server technology in which the user's request is fulfilled by running a script directly on a Web server to generate dynamic Web pages', i.e., a server-side script runs on the server rather than your computer.
- Applets are not given access to a local machine's file system. Thus, applet can neither read a file from the local machine, nor can it save a file on the local system.
- Transmission Control Protocol and Internet Protocol TCP/IP consist of a set of rules for exchanging information between applications, across networks.
- User Datagram Protocol (UDP) works as a datagram service between the clients and server. Datagram is a unit of information that consists of a frame and frame in turn consists of a set of bits.
- A port is a numeric address, which forms a common channel for communication of data using a particular protocol. A 16-bit number, ranging from 0 to 65535, represents ports.

NOTES

- A socket establishes an uninterrupted network connection between a client and a server using the port number. The server computer has a host name and a port number that the client computer uses to make a connection request.
- IPv4 addresses are uniquely used as identifiers, which work at network layer to identify the source or destination of IP packets.
- IP addresses may have fixed part and variable part depending upon the allocation of total addresses to an organization.
- Address is an identifier that is assigned to a device attached to a node in the Internet. It tells us about the source or destination of IP packets.
- Electronic mail or e-mail is one of the most popular applications of the Internet today. It refers to the process of transmission of messages in a communication network which was started by Ray Tomlinson in 1971.
- The exchange of messages between two terminals or hosts is done by using the SMTP along with software programs known as Mail Transfer Agents (MTAs).
- Message Transfer Agents (MTA) are programs that actually transfer mails. A system generally needs two MTAs—the client MTA to send mails and a server MTA to receive mails.
- The Internet is considered as hub of information as it provides vast amount of data, such as text, audio, video and other documents related to any topic or subject.
- Search engine improves the user's browsing experience. All those users who use Google search engine can get the quick and fast information.
- Remote Method Invocation (RMI) is a part of Java Development Kit (JDK). It allows us to develop distributed application. Distributed systems require computations that are running in different address spaces, particularly on different machines, to be able to communicate between one another.
- The RMI architecture is based on one important principle which states that the definition of behaviour and the implementation of that behaviour are separate concepts. RMI allows the code that defines the behaviour and the code that implements the behaviour to remain separate and to run on separate JVMs.
- The stub and skeleton layer of RMI lie just beneath the view of the Java developer. In this layer, the Proxy pattern, an object in one context is represented by another (the proxy) in a separate context.
- CORBA and Object Request Broker (ORB) clients and servers communicate with each other using IIOP.
- The RMI transport layer is designed to make a connection between clients and server, even in the face of networking obstacles.

3.12 ANSWERS TO 'CHECK YOUR PROGRESS'

1. JavaScript is an object oriented scripting language used to enable programmatic access to objects within both the client application and other applications.
2. The primary use of JavaScript is to write functions that are embedded or included from HTML pages and interact with the Document Object Model (DOM) of the page.

NOTES

3. VBScript is acronym of Visual Basic Scripting Edition and is referred as an Active Scripting language specifically developed by Microsoft modeled on Visual Basic.
4. VBScript is a scripting language or more specifically a scripting environment that can enhance HTML Web pages and makes them more active than a simple static display.
5. In VBScript, variables are used to hold values or expressions. Variables are named storage locations that can contain data that can be modified while script is running.
6. A scripting language or script language is referred as a programming language that supports the writing of program scripts.
7. ECMA-262 or ECMAScript is a standard for scripting languages published by the European Computer Manufacturer's Association. ECMA is a nonprofit international industry association that is engaged in the standardization of information and communications technologies.
8. Web browsers are applications for displaying Web pages.
9. PHP is a server side HTML embedded scripting language that you can use to create dynamic Web pages. It includes predefined functions that create, open, read, write and close files stored on the server. You can create XML-based Web applications using PHP.
10. Web pages are created using HTML files.
11. Applets are not given access to a local machine's file system. Thus, applet can neither read a file from the local machine, nor can it save a file on the local system.
12. Transmission Control Protocol/ Internet Protocol (TCP/IP) consists of TCP and UDP that are used to exchange the data over a network. The TCP/IP reserves the port number 1024.
13. The server socket is used for creating a server application, which enables a server to communicate with the client.
14. IPv4 addresses are uniquely used as identifiers, which work at network layer to identify the source or destination of IP packets.
15. Electronic mail or e-mail is one of the most popular applications of the Internet today. It refers to the process of transmission of messages in a communication network which was started by Ray Tomlinson in 1971.
16. Search engine improves the user's browsing experience. All those users who use Google search engine can get the quick and fast information.
17. Protocol handler is a feature of the Windows Search engine that is used to communicate with and enumerate the contents of stores, such as the file system, Messaging Application Program Interface (MAPI) e-mail database, and the CSC or offline files database. Like IFilters, protocol handlers are also extensible.
18. (a) VBScript, (b) <script>, (c) Function, (d) Web pages.
19. (a) False, (b) True, (c) True, (d) False.

3.13 QUESTIONS AND EXERCISES

NOTES

Short-Answer Questions

1. What is a JavaScript?
2. State the difference between Java and JavaScript.
3. Define the term VBScript.
4. What is URL encoding?
5. What is an `InputStream` class?
6. How are Web pages created?
7. State the limitations of applets.
8. What is UD protocol?
9. What is the code for implementing socket connection?
10. What is an electronic mail?

Long-Answer Questions

1. Are Java and JavaScript the same? Explain with the help of example.
2. Describe how JavaScript work.
3. Describe VBScript in HTML with the help of an example.
4. Discuss the VBScript data types.
5. What is an `OutputStream` class? Explain with the help of an example code.
6. What are the server-side scripting languages? Describe features of each.
7. Describe the applet class and the methods implemented in the applet class.
8. Explain the various constructors needed for creating the client socket.
9. How do you create a client program? Explain the code snippet.
10. What is IPv4 addressing? Classify them.
11. Write a note on subnetting for IP addresses.
12. What is user agent? Explain the various services provided by the UA.
13. Describe the various RMI architecture layers.

UNIT 4 INTERNET MARKUP LANGUAGES

NOTES

Structure

4.0 INTRODUCTION

In this unit, you will learn about the Internet markup languages, such as SGML and HTML. SGML is an ISO standard technology for generalized markup languages for documents. HTML, XHTML and XML are the examples of SGML based languages. HTML is a markup language which sets markup tags to describe Web pages. HTML markup tags are usually known as HTML tags which are keywords surrounded by angle brackets like <html>. HTML tags appear in pairs like and where the first tag in a pair is the start tag and the second tag is the end tag. The starting and ending tags are also called opening tags and closing tags. HTML elements can have attributes which provide additional information about an element. You will also learn about anchors. When you click on a link in an HTML page, an underlying <a> tag

NOTES

points to an address on the WWW. The HTML tables are defined with the `<table>` tag. A table is divided into rows with the `<tr>` tag and each row is divided into data cells with the `<td>` tag for table data which holds the content of a data cell. A `<td>` tag can contain text, links, images, lists, forms, other tables, etc. Finally, you will learn about forms and frames. A form can contain input elements like text fields, checkboxes and radio buttons, submit buttons and more. With frames, you can display more than one HTML document in the same browser window. The `<frame>` tag defines one particular window (frame) within a frameset. Each HTML document is known as a frame and each frame is independent of the others. Each frame element can hold a separate document. You will also learn about the significance of XHTML and DHTML. Internet Explorer is a series of graphical Web browsers, developed by Microsoft and included as a part of the Microsoft Windows operating systems. You will learn about the anatomy of Internet Explorer. Finally, you will learn about the netscape expensions, shockware and lingo.

4.1 UNIT OBJECTIVES

After going through this unit, you will be able to:

- Explain the significance of HTML
- Discuss various types of tags and attributes
- Differentiate between text styles and text arrangements
- Understand color and background of Web pages
- Explain lists and their types available in HTML
- Describe the attributes of `<image>` tag
- Explain the significance of hyperlink
- Understand the use of anchors
- Create table, frame, form and style sheet
- Explain the significance of XHTML and DHTML
- Convert a HTML document to XHTML
- Discuss the basic anatomy of Internet Explorer Web browser
- Describe the significance of shockwave and and lingo
- Discuss various netscape extensions

4.2 INTRODUCTION TO HTML AND SGML

In 1980, Tim Berners Lee, a physicist, now the Director of W3C (World Wide Web Consortium), designed a system to use and share documents. In 1989, Tim proposed a hypertext system based on the Internet and by end of 1991, wrote the first specifications for HyperText Markup Language or HTML which contained 20 elements. Thirteen of these elements still exist in HTML 4. HTML specifications are maintained by the W3C. The last HTML specification published by W3C was HTML 4.01.



HTML: A language that provides a very simple mechanism to format a Web page, outputting page in different formats, designing tables, embedding images and other objects

Advantages of HTML

HTML is a language that provides a very simple mechanism to format a Web page, outputting page in different formats, designing tables, embedding images and other objects. One can play with text according to its own requirement. Secondly, it is very

easy to learn and is the most widely accepted language for Web designing. One need not be an expert to write HTML documents.

HTML and World Wide Web

NOTES



WWW: A system of linked HTML documents over the Internet

The terms World Wide Web or WWW and Internet are used in everyday speech without any distinction. The Internet is a global system of computer systems interconnected to each other. Web is a service which runs on the Internet. **WWW** is a system of linked HTML documents over the Internet. This collection of information is linked by hypertext links and is known as a Web of information, that is available on the Internet and thus creates the WWW. A Web browser is used to view these pages on the Internet. HTML is the most widely accepted language of the Web. HTML is the language used for publishing on WWW. World Wide Web is primarily composed of HTML documents, which are transferred from the Web server to the Web Browser using the HTTP or HyperText Transfer Protocol. All Internet browsers follow HTML standards to view an HTML page. Extraction tools like Python follow HTML standards to extract information from Web pages.

Internet Protocols

HTTP is the application protocol which is used over the WWW to transfer HTML pages between your application (typically a Web browser) and the Web server publishing the HTML page. The standards for HTTP were developed by the W3C and IETF (Internet Engineering Task Force). The version of HTTP available in common use is HTTP/1.1.



HTTP: The standard for request/response based typical client-server computing

HTTP is the standard for request/response based typical client-server computing. A client is an application (a Web browser) and the server is the computer hosting the Web page. The client/application which submits an HTTP request to a server is referred as an *User Agent*. The Web Server which stores HTML files and resources is known as an *Origin Server*. HTTP uses Transmission Control Protocol/Internet Protocol or TCP/IP for reliable transport. The transfer of Web pages from server to client over HTTP is done on TCP/IP layer. HTTP is not constrained to use TCP/IP, it can use any Internet protocol that provides reliable transport. A Web server runs on an IP address and you can contact that Web server using the HTTP protocol and request a Web Page. A Web server then checks whether it hosts that page or not, if it does then it returns that page over HTTP and your browser displays the page.

Devices Used over the Internet



Proxies: Servers which act as a proxy for you over the network and hide the network behind it

Internet basically stands for Internetwork. On the Internet, different networks are interconnected via various Internet devices like routers, gateways, proxies, tunnels, etc. Routers and gateways are devices which connect two networks and do the job of routing a message from one network to another network. **Proxies** are servers which act as a proxy for you over the network and hide the network behind it. For HTTP protocol, this matrix is unknown as it relies on TCP/IP for the connection between two machines in different networks. HTTP can work over any protocol that provides a reliable transport. Discussing the Internet matrix is beyond the scope of this book.

Delivery Mechanism of HTML Document on World Wide Web

The HTML documents are delivered from a Web server to a client via the HTTP protocol running over TCP/IP. This HTTP transfer takes place in the following way:

- A Uniform Resource Locator or URL is typed in to the browser.

NOTES

- Server name portion from the URL is extracted and resolved into an IP Address using a global Internet database called the Domain Name System or DNS.
- This IP address is required to make a connection to the server over the TCP/IP layer. A transport connection is made to the server over the TCP/IP layer.
- A browser then requests the resource mentioned in the URL from the Web server over the HTTP protocol.
- If the resource is found, the server sends an OK response which then returns the HTML over the HTTP, otherwise it returns a 404 response.

File Naming Conventions in HTML

All HTML file use **.html** or **.htm** as file name extension. This extension uniquely identifies an HTML file and associates them directly to the Web browser.

Standard Generalized Markup Language

The **Standard Generalized Markup Language** (**ISO 8879:1986 SGML**) is an ISO (International Organization for Standards) standard technology for defining generalized markup languages for documents.

Generalized markup is based on the following two novel postulates:

1. Markup should be declarative. It should describe a document's structure and other attributes, rather than specify the processing to be performed on it. Declarative markup is less likely to conflict with unforeseen future processing needs and techniques.
2. Markup should be rigorous so that the techniques available for processing rigorously-defined objects like programs and databases can be used for processing documents as well.

HTML, XHTML and XML are all examples of SGML-based languages.

An SGML document may have three parts:

1. The SGML declaration.
2. The Prologue, containing a DOCTYPE declaration with the various *markup declarations* that together make a Document Type Definition (DTD).
3. The instance itself, containing one top-most element and its contents.

An SGML document may be composed from many entities (discrete pieces of text). In SGML, the entities and element types used in the document may be specified with a DTD, the different character sets, features, delimiter sets and keywords are specified in the SGML declaration to create the *concrete syntax o*f the document.

Although full SGML allows implicit markup and some other kinds of tags, the XML specification states the following features:

- Each XML document has both a logical and a physical structure.
- Physically, the document is composed of units called entities. An entity may refer to other entities to cause their inclusion in the document.
- A document begins in a 'root' or document entity. Logically, the document is composed of declarations, elements, comments, character references

and processing instructions all of which are indicated in the document by explicit markup.

NOTES

Optional Features

SGML generalizes and supports a wide range of markup languages as found in the mid 1980s. These ranged from terse wiki-like syntaxes to RTE-like bracketed languages to HTML-like matching-tag languages. SGML did this by a relatively simple default *reference concrete syntax* augmented with a large number of optional features that could be enabled in the SGML Declaration. Not every SGML parser can necessarily process every SGML document. Because each processor's *System Declaration* can be compared to the document's *SGML Declaration* it is always possible to know whether a document is supported by a particular processor.

Many SGML features relate to markup minimization. Other features relate to parallel asynchronous markup (CONCUR), to linking processing attributes (LINK) and to embedding SGML documents within SGML documents (SUBDOC).

The notion of customizable features was not appropriate for Web use, so one goal of XML was to minimize optional features. However XML's well-formed rules cannot support Wiki-like languages, leaving them un-standardized and difficult to integrate with non-text information systems. Actually, SGML is used to define a markup language. An example of an HTML document (one of the markup languages) follows:

```
<!DOCTYPE HTML"/strict.dtd">
    <HTML>
    <HEAD>
     <TITLE>My first HTML document</TITLE>
    </HEAD>
    <BODY>
   <P>Hello world!
   </BODY>
    </HTML>
```

The preceding HTML document is divided into a header (here, between `<HEAD>` and `</HEAD>`) and a body (here, between `<BODY>` and `</BODY>`). The title of the document is in the header, besides other information about the document. The content of the document is in the body. There is just one paragraph in it, markup with `<P>`.

Theoretically, every SGML document has both a logical and a physical structure. Logically, a document is made up of elements, declarations, attributes, character references, comments, and so on. All these are shown in the document by clear markup.

Physically, the document is made up of units known as entities. A document starts in a document entity. SGML is not only used for conventional document markup, but it can be used for marking up any type of text. Header, paragraphs, footnotes, sections, hypertext links, tables, images, etc., are the elements in a SGML text. Every element usually describes the following three parts:

(i) A start tag.
(ii) Content.
(iii) An end tag.

NOTES

The name of the element appears in the start tag (written <element-name>) and the end tag (written </element-name>). Elements may have related properties called attributes. The latter may have values (by default or set by authors or scripts). Attribute/value pairs show before the final “>” of an element’s start tag. In the start tag of an element numerous (legal) attribute value pairs, separated by spaces, may become visible. They can appear in any order. Numeric or symbolic names included in an SGML document are called character reference. These character references help in referring to rarely used characters or those that authoring tools make it difficult or impossible to enter. They begin with a ‘&’ sign and end with a semi-colon (;). Following are examples of character references are:

‘<’ correspond to the < sign.

‘>’ correspond to the > sign.

‘"’ correspond to the ′ mark.

‘å’ (in decimal) correspond to the letter ‘a’ with a small circle above it.

‘И’ (in decimal) correspond to the Cyrillic capital letter ‘I’.

SGML comments have the following syntax:

```
<!— this is a comment —>
<!— and so is this one,
which occupies more than one line —>
```

Between the markup declaration open delimiter (‘<!’) and the comment open delimiter (‘—’), white space is not allowed. However, it is allowed between the comment close delimiter (‘—’) and the markup declaration close delimiter (‘>’). To include a series of hyphens (‘—’) within a comment is a common error. Any information that shows between remarks has no particular meaning.

In any markup norm the angle brackets are used as start and end tag delimiters. However, in an SGML text, it is permissible to use other characters, provided an appropriate tangible syntax is defined in the text of the SGML declaration. For instance, an SGML interpreter may be programmed to parse GML markup, wherein the tags are delimited with a left colon and a right full stop, thus, an :e prefix indicates an end tag: :xmp.Hello, world:exmp.

As per the reference syntax, upper or lower case is not important in tag names, thus the three tags: (i) <quote>, (ii) <QUOTE> and (iii) <quOtE> are similar.

In SGML, tags could be substituted with delimiter strings, for example two equals signs (==) at the beginning of a line are the ‘heading start tag’ and two equals signs (==) after that are the ‘heading end-tag’. One characteristic of SGML is the presumptuous empty tagging, such that the empty end tag </> in <ITALICS>this</> takes its value from the closest preceding full start tag, which in this example is <ITALICS>. Thus, it closes the most recently opened item. The appearance is hence equal to <ITALICS>this</ITALICS>. SGML also permits implied markup, various types of tags and many other not obligatory features.

Every SGML parser does not automatically process every SGML text. However, as the system declaration of the processor can be contrasted to the SGML declaration of the text, it was for all time likely to understand if a text was supported by a particular processor or not. Parsing a SGML document that involves traversing the dynamically

retrieved entity graph, finding or imply tags and the element structure, and validating those tags against the grammar.

The SGML equivalent is known as Document Type Definition (DTD). It defines only the structure; DTD describes all the texts of a particular type, in terms of the tags that may be used to mark them. SGML without a DTD (for example, simple XML) is a grammar or a language; SGML with a DTD is a Meta language. A separate specification of style layout complementing the DTD is called style sheet. For each tag defined in the DTD, a style sheet provides a rule describing the way in which elements with the tag should be laid out. There may be more than one style sheet for a DTD, providing different appearance to the same structure. Since HTML tags were not enough for the class of Web pages developed over time there was a requirement from the Web designers to be able to define their own tags. Though SGML has that facility, it is not completely appropriate for use over the Internet. Work on making SGML compatible to the Internet led to the development of eXtensible Markup Language (XML) that gives all the facilities of SGML without the overhead (for complicated parsing) forced by SGML. In fact, XML permits Web designers to define their own DTDs for any type of document and Web pages are freed from the limitations of HTML's definition of a document.

4.2.1 HTML Tags

HTML pages are written in the form of HTML elements. HTML elements can be described as tags surrounded by angle 'brackets '<' and '>', which are interpreted by the browser to perform a specific formatting. Following is an example of how the basic HTML tags are used.

In HTML, every tag has a corresponding closing tag. A Web browser interprets these tags and then renders the display on the screen depending upon the tag and its attributes. Following are the features of an HTML file:

- The text between `<html>` and `</html>` describes the Web page.
- HEAD contains metadata information for HTML file.
- The text between `<body>` and `</body>` is the visible page content.
- All HTML tags should be properly ended by their corresponding closing tags otherwise it can interfere with the display and may not be as expected.
- All HTML tags are case insensitive.

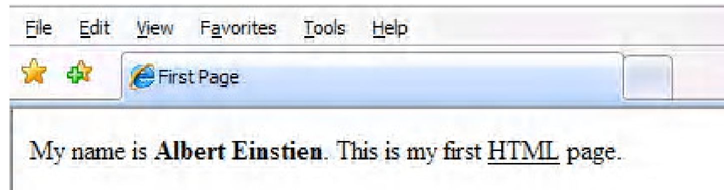
Creating an HTML Document

Try the following HTML in your browser:

```
<html>
<head>
     <title>First Page</title>
</head>
<body>
     My name is <b>Albert Einstien</b>. This
is my first <u>HTML</u> page.
</body>
</html>
```

NOTES

The page can then be displayed on your browser will be as follows:



In the above example, the content between the <body> tag is rendered on screen by browser. The text *Albert Einstien* surrounded by tag <b> tells the browser to display this text as bold. Similarly, the text *HTML* surrounded by tag <u> is displayed as underlined. The text *First Page* in the title tag, in the head section tells the browser to display it as the window title.

Essential Tags

Essential HTML tags are discussed below:

- **The <html> Tag:** The <html> tag is supported by all major browsers. This tag tells the browser that this is an HTML document. The <html> element is also known as the root element. The <html> tag is the container for all other HTML elements except for the <!DOCTYPE> tag.
- **HTML <body> Tag:** All text within the HTML document body will be displayed by the Web browser for the user to see. Therefore it is important to keep the body tags <body> and </body> around all of the content you want your visitors to see.

Deprecated Tags

Older HTML tags and attributes that have been superseded by other more functional or flexible alternatives whether as HTML or as Cascading Style Sheets or CSS are declared as deprecated in HTML 4 by the W3C that sets the HTML standards. Table 4.1 summarizes the list of Deprecated HTML tags.

Table 4.1 *Deprecated HTML Tags*

Deprecated	Description	Replacement
<applet>	Inserts applet	<object>
<basefont>	Sets font styles	Font style sheets
<center>	Centres elements	<div style="text-align:center">
<dir>	Directory list	
<font>	Applies font styles	Font style sheets
<isindex>	Adds search field	<form>
<menu>	Menu list	
<s>	Strikethrough	Text style sheets
<strike>	Strikethrough	Text style sheets
<u>	Underline	Text style sheets

4.3 TAGS AND ATTRIBUTES

HTML elements are tags and are enclosed within angle brackets '<' and '>', which act as indicators to a Web browser as to how the document is to be interpreted by the browser and ultimately presented on the user's computer screen. For example,

Check Your Progress

1. What is origin server?
2. What are proxies?
3. How HTML documents are delivered?
4. Name the file name extensions which are used in HTML file?
5. How HTML elements can be described?
6. Write the function of <head> and <body> parts of an HTML file.
7. Which element is known as root element?
8. Which deprecated tag is used as <object> replacement to insert applets in HTML?

NOTES

'Albert Einstien' is an HTML element which tells the browser to display the text placed between the shorting and closing tag as bold. Even the plain text is an HTML element. Elements may represent links, paragraphs, headings or some embedded objects like audio or video. HTML elements may contain other elements which may be empty. Empty elements consist of only a single tag, with or without any attributes, appended by the forward slash '/'.

Following are the few basic characteristics of HTML tags:

- An HTML element starts with a start tag/opening tag.
- An HTML element ends with an end tag/closing tag.
- The element content refers to everything between the start and the end tag.
- Some HTML elements have empty content.
- Empty elements are closed in the start tag, for example,
 and <hr/> are empty elements closed in a single tag.
- Most HTML elements can have attributes.

All HTML elements are SGML (Standard Generalized Markup Language) elements which meet the specifications of various HTML Document Type Definition or DTDs'. HTML elements are defined by many open standards, initially by IETF (Internet Engineering Task Force) and now by W3C.

Primary function of HTML is to add semantic information and structural view to the raw text of the document.

***Notes*:** (i) An HTML element should always have its closing tag, otherwise formatting on screen could be different from what expected.

(ii) HTML tags are not case sensitive. The World Wide Web Consortium or W3C recommends lowercase in HTML 4 and demands lowercase tags in the future versions of XHTML.

HTML Source

The content written between the <html> tags is known as an HTML source and is interpreted by the browser to render the display. You can go to the browser window and right click the mouse. It will pop-up a menu. There will be a menu item 'View Source', select this option. A text editor window will open showing the content of HTML file which you opened in browser.

HTML Tags and Arrangement

In HTML, only one rule follow, '*every opening tag should have a closing tag*'. *If tags are not closed properly*, then the display on your browser screen could be different from what is expected.

In general practice, use some indentation for every child element with respect to its parent element. Use a new line with proper indentation whenever you start a new element. It keeps the source viewable, and easy to read and understand.

Rule for Whitespace

In HTML, extra white spaces and new lines are neglected. Any number of lines or spaces is count as one space. Try the following HTML in your browser:

```
<html>
<head>
<title>First
Page</title>
```

NOTES

```
   </head>
   <body>
   Hi everybody,this is an example of nesting of HTML
elements.
    <b>This a text line in bold with  parts in
   <i>Italic text</i>            and
   <u>Underlined Text</u></b>.
   </body>
   </html>
```

To add extra space, a special whitespace character **` `** is used in HTML and to add a new line, the `<br/>` tag is used.

Comments for Better Understanding

One can also insert comments into the HTML file and these comments will not display on the browser. HTML comments are written in between `<!–` and `–>` tags. Try the following HTML below in your browser:

```
<html>
<head>
     <title>First  Page</title>
</head>
<body>
<!– This is a HTML Comment and will not be
displayed on the screen   –>
     Hi everybody, this is an example of
     nesting of HTML elements. <b>This a text
     line in bold with parts in <i>Italic
     text</i>   and <u>Underlined Text</u></
     b>.
</body>
</html>
```

You will not see the commented line on the browser. It is neglected by the browser. You can verify it by viewing the source of the displayed HTML. Right-click mouse in the browser window and select 'View Source' from the pop-up menu.

Special Characters

In HTML, few characters have a special meaning like '<', '>', '&', etc. Angle brackets represents the start or end of an HTML tag. Ampersand (&) represents the start of special characters. To render their display on the screen, HTML uses special characters to represent them.

Table 4.2 shows a list of few commonly used HTML special characters.

NOTES

Table 4.2 *List of HTML Special Characters*

Character	Entity Name	Description
&	&	Ampersand
¢	¢	Cent
©	©	Copyright
÷	÷	Divide
>	>	Greater than
<	<	Less than
µ	µ	Micron
·	·	Middle dot
¶	¶	Pilcrow (paragraph sign)
±	±	Plus/Minus
€	€	Euro
£	£	British Pound Sterling
®	®	Registered
§	§	Section
™	™	Trademark
¥	¥	Japanese Yen

4.3.1 <body> Element

The body element defines the page body. The <body> element contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc.

Table 4.3 shows the attributes used in the <body> tag.

Table 4.3 *Attributes of the* <body> *Tag*

Attribute	Value	Description
alink	rgb(x,x,x) #xxxxxx colorname	Specifies the colour of an active link in a document.
background	URL	Specifies a background image for a document.
bgcolor	rgb(x,x,x) #xxxxxx colorname	Specifies the background color of a document.
link	rgb(x,x,x) #xxxxxx colorname	Specifies the default color of unvisited links in a document.
text	rgb(x,x,x) #xxxxxx colorname	Specifies the color of the text in a document.
vlink	rgb(x,x,x) #xxxxxx colorname	Specifies the color of the visited links in a document.
class	Classname	Specifies a classname for an element.
dir	rtl ltr	Specifies the text direction for the content in an element.
id	Id	Specifies a unique id for an element.
lang	language_code	Specifies a language code for the content in an element.
style	style_definition	Specifies an in-line style for an element.
title	Text	Specifies extra information about an element.
xml:lang	language_code	Specifies a language code for the content in an element, in XHTML documents.

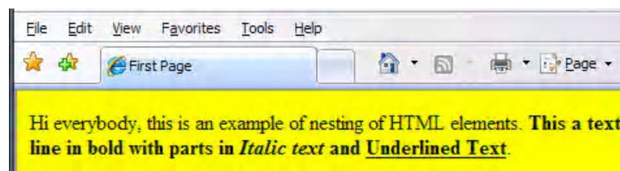
NOTES

bgcolor

The attribute **bgcolor** is used to specify the background color of the Web document when it is displayed in the browser. Try the following HTML in your browser:

```
<html>
<head>
     <title>First  Page</title>
</head>
<body  bgcolor='yellow'>
     Hi  everybody,  this  is  an  example  of
     nesting  of  HTML  elements.  <b>This  a  text
     line  in  bold  with  parts  in  <i>Italic
     text</i>   and  <u>Underlined  Text</u></
     b>.
</body>
</html>
```

This will set the background color of the page displayed as yellow. Try changing this value to green or blue and refresh your page. The output on the screen will be displayed as follows:



link

Attribute **link** is an attribute of the <body> tag. The **link** attribute specifies the default color of unvisited links in a document. The value of this attribute is a color value specified either by its standard name or by its RGB value. Color value could be a color name likes red, blue, green, yellow, etc., or any standard color name. It could be an RGB value specified in hexadecimal number format like green value is `00ff00`, red is `ff00000` and blue is `00000ff.`

alink

Attribute **alink** is an attribute of the <body> tag. The value of this attribute is a color value specified either by its standard name or by its RGB value. It renders all the links present in the Web page of the specified color value. Color value could be a color name like red, blue, green, yellow, magenta, etc., or any standard color name. It could be an Red Green Blue or RGB value as specified in a hexadecimal number format like green value is `00ff00`, red is `ff0000` and blue is `0000ff`.

vlink

The attribute **vlink** is an attribute of the <body> tag. The value of this attribute is a color value as specified either by its standard name or by its RGB value. It renders all the visited links present in the Web page of the specified color value. Color value could be a color name like red, blue, green, yellow, magenta, etc., any standard color name. It could be an RGB value specified in hexadecimal number format like green value is `00ff00`, red is `ff0000` and blue is `0000ff`.

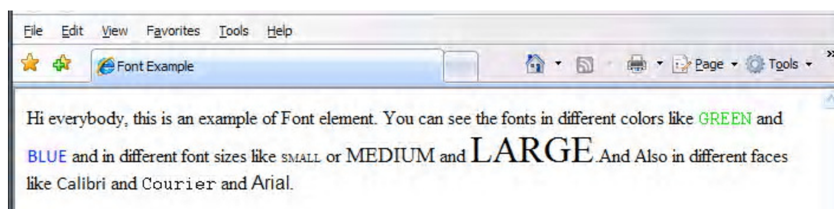
NOTES

4.3.2 <font> Element

The <font> tag specifies the face, color and size of the text written in this tag. The element <font> is used to format the text into the desired font size and color. Try the following HTML in your browser:

```
<html>
<head>
    <title>Font Example</title>
</head>
<body>
Hi everybody, this is an example of Font element. You can
see the fonts in different colors like <font color='0x00ff00"
face='Courier'>GREEN</font> and <font color='0x0000ff'
face='Calibri'>BLUE</font> and in different font sizes like
<font size='1">SMALL</font> or <font size='4">MEDIUM</font>
and <font size='6">LARGE</font>.And Also in different faces
like <font face='Calibri'>Calibri</font> and <font
face='Courier'>Courier</font> and <font face='Arial'>Arial</
font>.
</body>
</html>
```

The output on the screen will be display as follows:



Attributes associated with the <font> tag have been described in Table 4.4.

Table 4.4 *Attributes of the* <font> *Tag*

Attribute	Value	Description
color	rgb(x,x,x) #xxxxxx colour name	Specifies the color of text.
face	font_family	Specifies the font of text.
size	number	Specifies the size of text.
class	classname	Specifies a classname for an element.
dir	rtl ltr	Specifies the text direction for the content in an element.
id	Id	Specifies a unique id for an element.
lang	language_code	Specifies a language code for the content in an element.
style	style_definition	Specifies an inline style for an element.
title	Text	Specifies extra information about an element.

***Note*:** The <font> tag is deprecated in HTML 4.01 specs.

color Attribute

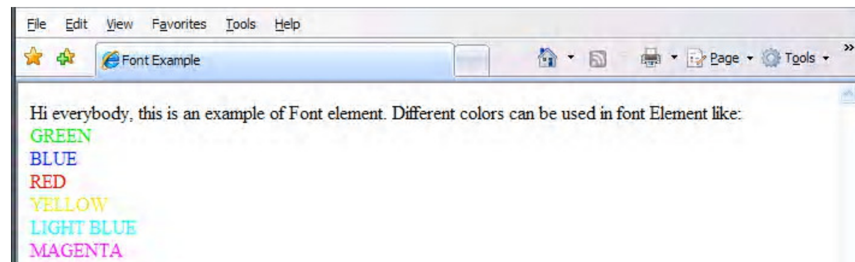
In the above example, the **color** attribute specified in the <font> tag renders the text in that color. Color value could be a color name like red, blue, green, yellow, magenta, etc., or any standard color name. It could be an RGB value specified in

NOTES

hexadecimal number format like green value is `00ff00`, red is `ff0000` and blue is `0000ff`. Try the following HTML in your browser:

```
<html>
<head>
    <title>Font Example</title>
</head>
<body>
Hi everybody, this is an example of Font element. Different
colors can be used in font Element like:
<br/> <font color='00ff00" >GREEN</font>
<br/><font color='0000ff' >BLUE</font>
<br/><font color='ff0000" >RED</font>
<br/><font color='ffff00" >YELLOW</font>
<br/><font color='00ffff' >LIGHT BLUE</font>
<br/><font color='ff00ff' >MAGENTA</font>
</body>
</html>
```

The output on the screen will be displayed as follows:



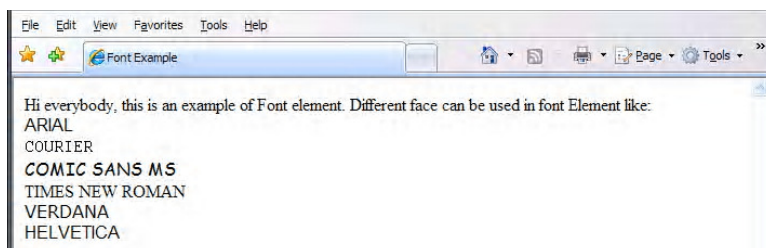
`face` Attribute

The **`face`** attribute specified in the `<font>` tag renders the text in that font face. The `face` value is the name of the font like `courier`, `calibri`, `arial`, etc. The font used for text formatting needed to be installed on the client machine otherwise the browser will not be able render the text in that font and will use the browser default font instead. Try the following HTML in you browser:

```
<html>
<head>
    <title>Font Example</title>
</head>
<body>
Hi everybody, this is an example of Font element. Different
face can be used in font Element like:
<br/> <font face='arial' >ARIAL</font>
<br/><font face='courier' >COURIER</font>
<br/><font face='comic sans ms' >COMIC SANS MS</font>
<br/><font face='times new roman' >TIMES NEW ROMAN</font>
<br/> <font face='verdana' >VERDANA</font>
<br/> <font face='helvetica' >HELVETICA</font>
```

```
</body>
</html>
```

The output on the screen will be displayed as follows:



NOTES

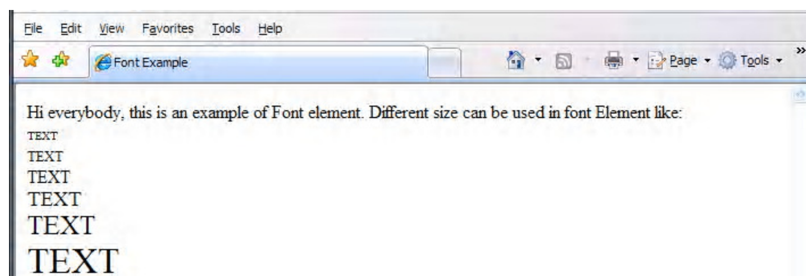
***Note*:** Most browsers prefer the font names in lower case.

`size` Attribute

The **`size`** attribute specified in the `<font>` tag renders the text in that particular font size. The `size` value is an integer number that specifies the font size in pixel point. Try the following HTML in your browser:

```
<html>
<head>
    <title>Font Example</title>
</head>
<body>
Hi everybody, this is an example of Font element. Different
size can be used in font Element like:
<br/> <font size='1" >TEXT</font>
<br/> <font size='2" >TEXT</font>
<br/> <font size='3" >TEXT</font>
<br/> <font size='4" >TEXT</font>
<br/> <font size='5" >TEXT</font>
<br/> <font size='6" >TEXT</font>
</body>
</html>
```

The output on the screen will be displayed as follows:



4.4 TEXT STYLES AND TEXT ARRANGEMENTS

Text is also an HTML element. Text formatting is the major crux of HTML. Any text other than tags and attributes, which is present in the `<body>` tag of HTML file will be displayed on the screen. Later on in this book, you will learn various HTML tags for formatting text and effects.

NOTES

Some other text formatting tags are as follows in Table 4.5.

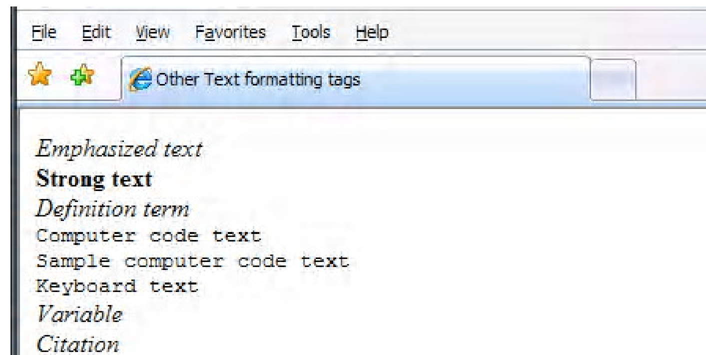
Table 4.5 *Text Formatting Tags*

Tag	Description
<em>	Renders as emphasized text.
<strong>	Renders as strong emphasized text.
<dfn>	Defines a definition term.
<code>	Defines computer code text.
<samp>	Defines sample computer code.
<kbd>	Defines keyboard text.
<var>	Defines a variable part of a text.
<cite>	Defines a citation.

Try the following HTML in your browser:

```
<html>
<head>
    <title>Other Text formatting tags</title>
</head>
<body>
    <em>Emphasized text</em></br>
    <strong>Strong text</strong></br>
    <dfn>Definition term</dfn></br>
    <code>Computer code text</code></br>
    <samp>Sample computer code text</samp></br>
    <kbd>Keyboard text</kbd></br>
    <var>Variable</var></br>
    <cite>Citation</cite>
</body>
</html>
```

The output on the screen will be displayed as follows:



Font Style Tags

The font style tags used in HTML have been summarized in Table 4.6.

Table 4.6 *Font Style Tags*

Tag	Description
<i>	Renders as italic text.
<b>	Renders as bold text.
<big>	Renders as bigger text.
<small>	Renders as smaller text.
<tt>	Renders as tele type text.

NOTES

Text Scripting Tags

Tags which are used to script text are as follows:

- **<sub>:** The <sub> tag defines subscript text. Subscript text appears half a character below the baseline. Subscript text can be used for chemical formulas like O_2.
- **<super>:** The <sup> tag defines superscript text. Superscript text appears half a character above the baseline. Superscript text can be used for footnotes, like WWW[1].

Element Grouping Tags

Tags which are used to group similar kinds of elements are as follows:

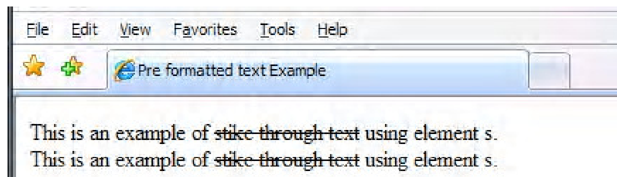
- **<fieldset>:** This tag is used to logically group elements together. It draws a box around the related elements.
- **<legend>:** This tag provides a caption for <fieldset> elements.

Striking Text

The tags <s> and <strike> are used to strike through the text. Try the following HTML in your browser:

```
<html>
<head>
    <title>Pre formatted text Example</title>
</head>
<body>
    This is an example of <s>stike through text</s> using
element s. <br/>
    This is an example of <strike>stike through text</strike>
using element strike. <br/>
</body>
</html>
```

The output on the screen will be displayed as follows:



Text Editing Tags

Tags which are used to format the edited text in the HTML document. They are as follows:

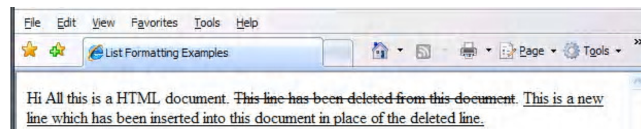
- **<del>**: Defines text that has been deleted from a document.
- **<ins>**: Defines text that has been inserted into a document.

NOTES

Try the following HTML in your browser:

```
<html>
<head>
    <title>List Formatting Examples</title>
</head>
<body>
    <p>
Hi All this is a HTML document. <del>This line has been
deleted from this document</del>. <ins>This is a new line
which has been inserted into this document in place of the
deleted line</ins>.
    </p>
</body>
</html>
```

The output on the screen will be displayed as follows:



Text Properties

These properties are used to set the behaviour of the text on the screen. Table 4.7 lists the text properties as follows:

Table 4.7 *Text Properties*

Property	**Description**
color	Sets the color of text.
direction	Specifies the text direction/writing direction.
letter-spacing	Increases or decreases the space between characters in a text.
line-height	Sets the line height.
text-align	Specifies the horizontal alignment of text.
text-decoration	Specifies the decoration to be added to the text.
text-indent	Specifies the indentation of the first line in a text block.
text-shadow	Specifies the shadow effect added to text.
text-transform	Controls the capitalization of text.
vertical-align	Sets the vertical alignment of an element.
white-space	Specifies how the white space inside an element is handled.
word-spacing	Increases or decreases the space between words in a text.

Background Properties

These properties when specified in the element, change the background of the element. These properties can be a background image or background color, etc. Table 4.8 shows a list of these properties as follows:

NOTES

Table 4.8 Background Properties as used in HTML

Property	Description
Background	Sets all the background properties in one declaration.
Background-attachment	Sets whether a background image is to be fixed or scrolled with the rest of the page.
Background-color	Sets the background color of an element.
Background-image	Sets the background image for an element.
Background-position	Sets the starting position of a background image.
Background-repeat	Sets how a background image will be repeated.

4.5 LISTS AND THEIR TYPES

Following are the types of HTML lists:

- **HTML Unordered Lists**: An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag. The list items are marked with bullets typically small black circles.
- **HTML Ordered Lists**: An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag. The list items are marked with numbers.
- **HTML Definition Lists***:* A definition list is a list of items, with a description of each item. The `<dl>` tag defines a definition list. The `<dl>` tag is used in conjunction with `<dt>` (defines the item in the list) and `<dd>` (describes the item in the list).

List Formatting Tags

The list formatting tags used in HTML have been shown in Table 4.9.

Table 4.9 List Formatting Tags

Tag	Description
	Creates an unordered List.
	Creates an ordered List.
	Defines a list item.
<menu>	Defines a menu list.
<dir>	Defines a directory list.

In HTML, you can define a definition list. A definition list is a list of elements with their respective definitions. Tags which are used in defining a definition list are as follows:

- **<dl>**: Defines a definition list.
- **<dt>**: Defines an item in definition list.
- **<dd>** **:** Describes the item in the list.

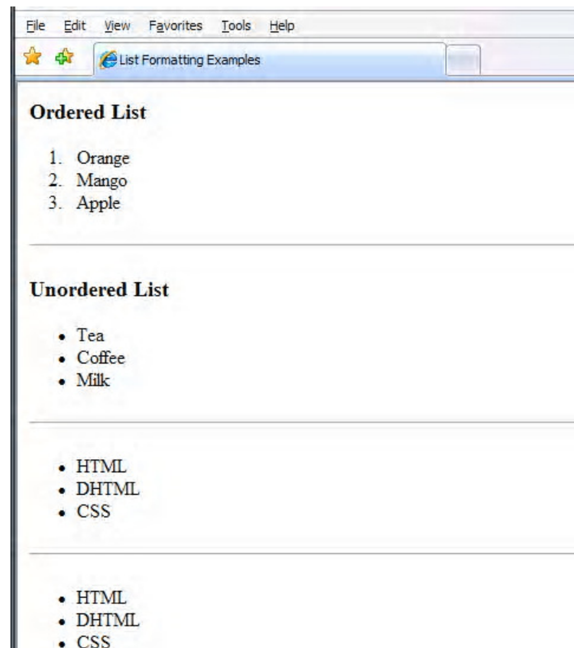
Try the following HTML in your browser:

```
<html>
<head>
    <title>List Formatting Examples</title>
```

NOTES

```
</head>
<body>
    <h3>Ordered List</h3>
    <ol>
       <li>Orange</li>
       <li>Mango</li>
       <li>Apple</li>
    </ol>
<hr/>
    <h3>Unordered List</h3>
<ul>
       <li>Tea</li>
       <li>Coffee</li>
       <li>Milk</li>
    </ul>
    <hr/>
<menu>
       <li>HTML</li>
       <li>DHTML</li>
       <li>CSS</li>
    </menu>
    <hr/>
<dir>
       <li>HTML</li>
       <li>DHTML</li>
       <li>CSS</li>
    </dir>
</body>
</html>
```

The output on the screen will be displayed as follows:



Check Your Progress

9. Write one of the features of HTML tags.
10. Why attribute `bgcolor` is used?
11. Name the tag which is used to insert a hyperlink in an HTML page.
12. Why `<face>` attribute is specified in the `<font>` tag?
13. What does `<size>` attribute specified in the `<font>` tag renders?

4.6 ATTRIBUTES OF IMAGE TAG

NOTES

In HTML, images are defined with the <img> tag. Basically, the <img> tag is empty, which means that it contains attributes only and has no closing tag. To display an image on a page, you need to use the src attribute, where src stands for 'source'. The height and width attributes are used to specify the height and width of an image. Hence, both the height and width attributes for an image must be specified. If these attributes are set, the space required for the image is reserved when the page is loaded. In these attributes are not defined then the browser does not know the size of the image. The effect will be that the page layout will change while the images load. When a Web page is loaded, it is the browser that gets the image from a Web server and inserts it into the page. Therefore, the images must be stored at specified link location otherwise you will get a broken link icon. The broken link icon is shown if the browser cannot find the image.

4.6.1 Tag for Images

Images used in HTML documents are classified into the following two classes:

1. Inline images.
2. External images.

Inline images are seen on a Web page with the text and links. They are loaded automatically when an HTML document is viewed on the Web browser. External images cannot be displayed directly. They are stored isolated from the HTML document and are loaded only when asked for Graphical Interchange Format (GIF) and Joint Photographic Experts Group (JPEG) are the generally used formats for images.

Using the <img> Tag

The <img> tag can be used to display the inline images. The following program shows the syntax to implement the <img> tag:

```
<img
    src="location"
    height="height"
    width="width"
    name="imageName"
    alt="alterntiveText"
    align="alignment"
    border="borderWidth"
    hspace="horizMargin"
    vspace="verticalMargin"
    usemap="#mapName"
>
```

In the preceding program, the various attributes of the <img> tag are as follows:

- **src:** It specifies the location of the image that needs to be inserted into an HTML document.
- **height:** It specifies the height of the image either in pixels or percentage height with respect to the Web browser window.
- **width:** It specifies the width of the image either in pixels or percentage width with respect to the Web browser window.

NOTES

- **name:** It refers to the name for the image.
- **alt:** It specifies the text to be displayed when the Web browser does not support the `<img>` tag.
- **align:** It specifies the image alignment in reference to the surrounding text. The values that can be assigned to this attribute are as follows:
  - o **right:** It is used to align an image according to the right margin.
  - o **left:** It is used to align an image according to the left margin.
  - o **top:** It is used to align an image according to the top margin of the current line.
  - o **absmiddle:** It is used to align the centre an image according to the middle of the current line.
  - o **absbottom:** It is used to align the middle of an image according to the bottom of the current line.
  - o **texttop:** It is used to align the top of an image according to the top of the current line.
  - o **middle:** It is used to align the middle of the image according to the text baseline of the current line.
  - o **baseline:** It is used to align the bottom of an image according to the text baseline of the current line.
  - o **bottom:** It is used to align the image in the same way as the BASELINE attribute. This is the default value of the align attribute.
- **border:** It specifies thickness of the border of an image, in pixels.
- **hspace:** It specifies the margin in pixels between the surrounding text and the right and left edges of the image.
- **vspace:** It specifies the margin in pixels between the surrounding text and the top and bottom edges of the image.
- **usemap:** It represents an image map and specifies which MAP tag is used to define the mapping for this image.

The following code shows how to use the <img> tag:

```
<img src="image1.jpg" height=300 width=200 border="2">
```

In the preceding program, the <img> tag is utilized to insert image1.jpg in an HTML document. The width of the image is specified as 200 and the height is specified as 300. The thickness of the border is specified as 2 pixels.

You can also wrap an image with text around it. The following code shows how to insert an image that is surrounded by text:

```
<p>
<img src="fig.jpg" hspace=40 border=2 align=LEFT >
This text will appear with the image.
</p>
```

In the preceding program, the <p> tag displays both image and text. The <img> tag can be used to insert a left-aligned image. The text, 'This text will appear with the image', will be displayed to the right side of the image with a margin of 40 pixels.

Using Images as Hyperlink

Images can also work as hyperlinks. You have to use the <img> tag inside the opening and closing part of the anchor tag. The following code depicts shows how to use an image as a hyperlink:

```
<a href="index.html">
 <img src="fig.jpg">
</a>
```

In the preceding code, the <img> tag is nested inside the anchor tag. The image works as the hyperlink that links the page, `index.html`. The Web browser shows the HTML document, `index.html` when you click on the image, `fig.jpg`.

Text and image can be used together to display a hyperlink. The following code shows how to use text and image together to represent a hyperlink:

```
<html>
<head>
</head>
<body>
<a href="index.html"><IMG src="button.jpg">Click here to view
index.html.</a>
</body>
</html>
```

In the preceding code, the image, `button.jpg`, and the text, `Click here to view index.html`, points to the same HTML document, index.html. Save the preceding code as Hyperlinks.html. The following screen shows the output when `Hyperlinks.html` is viewed in the MSIE Web browser.



4.7 HYPERLINK

A **hyperlink** or link is a reference to data that the reader can directly follow or that is followed automatically. It points to a whole document or to a specific element within a document. Hypertext is text which is used with hyperlinks. A software system is used to view and create hypertext called as hypertext system and also to create a hyperlink for the Web sites. A hyperlink has an anchor which is the location within a document from which the hyperlink can be followed, the document containing a hyperlink is known as its source document. Hyperlinks are often used to implement reference mechanisms, such as tables of contents, footnotes, bibliographies, indexes, letters and glossaries.

Links appeared on Web page that navigates the corresponding Web pages as a reference. If user navigates the pages to search or get the information through links, that is also called in Web technology as hyperlink, provides a very quick journey of

NOTES



Hyperlink: It is a reference to data that the reader can directly follow or that is followed automatically

NOTES

Web pages. Link is same as citation in literature. It is combined with suitable access protocol and data networking. The syntax of creating link is as follows:

```
<a  href="url">Link  text</a>
```

The start tag contains the attributes of specified link whereas 'Link text' is declared as the highlighted text that is displayed on Web page as link. The element content could be text or graphics, for example buttons, etc. You would be able to link from an image or other HTML element. Creating link is possible by writing the HTML code as follows:

```
<html>
<body>
<p>
<a href="feedback.htm">Feedback</a>
</p>
<p>
<a href="http://www.abc.com/">Link to World Wide Web</a>
</p>
</body>
</html>
```

Result of the above Web code is as follows that provides the following link:

Feedback

Link to World Wide Web

After clicking on **Feedback** link and **Link to World Wide Web** you can navigate on Feedback page and referenced Web site. If you set `target="_blank"` on the coding, the link will open a new window where referenced information resides.

So, the proper coding is written as follows:

```
<a href="feedback.htm" target="_blank" >Feedback</a>
```

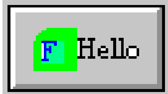
The HTML anchor is defined by `<a>`, which is used to define both anchors and hyperlinks. The `href` attribute is set with `<a>` element that defines the link as marked with underline and generally highlighted with blue color.

Feedback

Hyperlinks point to resources that appear on the Web page. These resources can be an HTML document, an image, a movie or a sound file, etc. The following HTML code shows how an image is connected with link:

```
<html>
<body>
<p>
<a href="feedback.htm"><img border="0" src="button_def.gif"
width ="65" height="38">
</a>
</p>
</body>
</html>
```

The output on the screen will be displayed as follows:



NOTES

Other attributes used with tag **<a>** are shown in Table 4.10.

Table 4.10 *Attributes of Tag* <a>

Attribute	Value	Description
charset	char_encoding	Specifies the character set of a linked document.
coords	coordinates	Specifies the coordinates of a link.
href	URL	Specifies the destination of a link.
hreflang	language_code	Specifies the language of a linked document.
name	section_name	Specifies the name of an anchor.
rel	Text	Specifies the relationship between the current document and the linked document.
rev	Text	Specifies the relationship between the linked document and the current document.
shape	default rect circle poly	Specifies the shape of a link.
target	_blank _parent _self _top framename	Specifies where to open the linked document.

Link on Same Page

Linking on same page is basically created, if the Web document is too long and user navigates the pages quickly, for example, chapter wise navigation on the Web page or e-book. Linking on same Web page is written as follows:

```
<html>
<body>
<p>
<a href=”C4">You can navigate on Chapter 4 from here.></a>
</p>
<h2>Chapter 1</h2>
<p>This chapter explains MS Word.</p>
<h2>Chapter 2</h2>
<p>This chapter explains MS Excel.</p>
<h2>Chapter 3</h2>
<p>This chapter explains MS PowerPoint.</p>
<h2><a name=”C4">Chapter 4</a></h2>
<p>This chapter explains a detailed description about MS
Access. </p>
</body>
</html>
```

The output on the screen will be displayed as follows:

You can navigate on Chapter 4 from here.

NOTES

Chapter 1

This chapter explains MS Word.

Chapter 2

This chapter explains MS Excel.

Chapter 3

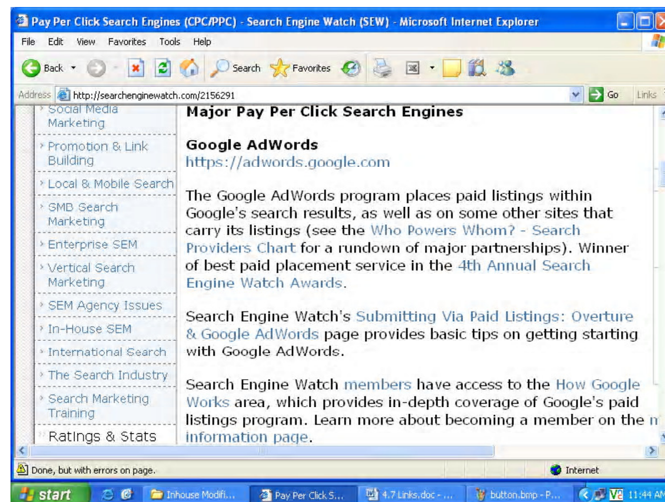
This chapter explains MS PowerPoint.

Chapter 4

This chapter explains a detailed description about MS Access.

In the above result, if you click on '**You can navigate on Chapter 4 from here**'. marked hyperlink, you will get the Chapter 4 page quickly and directly.

In these days, links are useful to search engine optimization. The screen shows how links are useful for major Pay Per Click or (PPC) search engines. These search engines provide various links that make Web page more powerful.



In each search engine index, titled links create the popularity of your Web site by PPC or CPC mechanisms. Users browse the sites that give the quality and volume of the pages. You can view the referring traffic of the Web pages. Commercial organizations make powerful linking and tagged with search engine optimization to make their strategies about their business growth.

Note: The broken link also known as a dangling link or a dead link that takes place when the Web server responds but when the specific page is not found. The most common broken link is known as a 404 error representing a broken link.

4.7.1 Links to External Documents

Internal link is used to create a link to another page on your Web site using <a> tag as follows:

```
<A HREF="internal.htm">Internal Link</a>
```

The result appears in the following way:

Internal Link

This link is also known as relative link.

External link is used to create a link to another page on another Web site using `<a>` tag as follows:

```
<A HREF="http://www.google.com/">External Link</a>
```

The result appears in the following way:

External Link

The external link is also called an absolute link.

NOTES

4.8 ANCHORS

An **anchor** represents a string or text which marks the beginning and/or the end of a hypertext link. The text between the opening tag and the closing tag is either the start or destination (or both) of a link. Attributes of the anchor tag are as follows.

- `href`: If the `href` attribute is present, the anchor is sensitive text and it shows the start of a link, for example `<a href="#identifier"> CLICK HERE...</a>`.
- `name`:This attribute allows the anchor to be the destination of a link. The value of the attribute is an identifier for the anchor, for example `<a name= "identifier">...TO JUMP HERE</a>`

All other attributes, such as `rel` and `ref` attributes are optional although one of `name` and `href` is necessary for the anchor to be useful. The `<a>` tag defines an anchor. An anchor can be used in the following ways:

- To create a link to another document, use the `href` attribute.
- To create a bookmark inside a document, use the `name` attribute.

The `<a>` element is usually referred as a link or a hyperlink. The most important attribute of the `<a>` element is the `href` attribute which indicates the link's destination. By default, links appear as follows in all browsers:

- An unvisited link is underlined and blue.
- A visited link is underlined and purple.
- An active link is underlined and red.

Following code is used to represent an anchor representing Microsoft.com:

```
<a href="http://www.microsoft.com">Visit Microsoft.com!</a>
```

Table 4.11 summarizes the event attributes used with `<a>` tag.



Anchor: A string or text which marks the beginning and/or the end of a hypertext link

NOTES

Table 4.11 Event Attributes of <a> Tag

Attribute	Value	Description
onblur	Script	Script is run when an element loses focus.
onclick	Script	Script is run on a mouse click.
ondblclick	Script	Script is run on a mouse double click.
onfocus	Script	Script is run when an element gets focus.
onmousedown	Script	Script is run when mouse button is pressed.
onmousemove	Script	Script is run when mouse pointer moves.
onmouseout	Script	Script is run when mouse pointer moves out of an element.
onmouseover	Script	Script is run when mouse pointer moves over an element.
onmouseup	Script	Script is run when mouse button is released.
onkeydown	Script	Script is run when a key is pressed.
onkeypress	Script	Script is run when a key is pressed and released
onkeyup	script	Script is run when a key is released.

4.9 CREATING TABLE

The **<table>** tag is used to define an HTML table. An HTML table is divided into rows which in turn are divided into columns. The use of an HTML table is to format the data into rows and columns. Try the following HTML in your browser:

The tags used to format a table have been shown in Table 4.12.

Table 4.12 Table Formatting Tags

Tag	Description
<thead>	The <thead> tag is used to group the header content in an HTML table.
<tbody>	The <tbody> element is used to group the body content in an HTML table
<tfoot>	The <tfoot> element is used to group the footer content in an HTML table.



The <table> tag: It is used to define an HTML table

Advantages of Tables

HTML tables are used to format text into the row and column format. Not only the text but also other HTML controls can be formatted using HTML tables.

Limitations of Table

In HTML tables, you cannot change the width of columns in each row. The size that is mentioned in the first row is what the table follows for column size rendering. Another limitation of an HTML table is that if size of content varies then the table cell is expanded in length or width depending on the wrapping attribute used thus not displaying the expected rendering.

4.9.1 Attributes of Table Elements

Attributes which are supported in the tag <table> are listed in Table 4.13.

Check Your Progress

14. In HTML, how images are defined?
15. Differentiate between inline and external images.
16. What does align specify?
17. Define the term hyperlink.
18. What does an anchor represent?

NOTES

Table 4.13 *Attributes of the* `<table>` *Tag*

Attribute	Value	Description
Align	left center right	Specifies the alignment of a table according to the surrounding text.
Bgcolor	rgb(x,x,x) #xxxxxx colorname	Specifies the background color for a table.
Border	Pixels	Specifies the width of the borders around a table.
cellpadding	Pixels	Specifies the space between the cell wall and the cell content.
cellspacing	Pixels	Specifies the space between cells.
frame	void above below hsides lhs rhs vsides box border	Specifies which parts of the outside borders should be visible.
Rules	none groups rows cols all	Specifies which parts of the inside borders that should be visible.
Summary	Text	Specifies a summary of the content of a table.
Width	pixels%	Specifies the width of a table.
Class	Classname	Specifies a classname for an element.
Dir	rtl ltr	Specifies the text direction for the content in an element.
Id	Id	Specifies a unique id for an element.
Lang	language_code	Specifies a language code for the content in an element.
Style	style_definition	Specifies an inline style for an element.
Title	Text	Specifies extra information about an element.
xml:lang	language_code	Specifies a language code for the content in an element in XHTML documents.

`align` Attribute

The **`align`** attribute is used to align a table to right or left or center into the Web page. It sets the table in that particular location on the screen.

Changing the value of the `align` attribute to '`right`' from '`center`' will change the table alignment to the right of the browser window. By default, the alignment of the table is towards `left`.

`width` Attribute

The attribute `width` is used to set the width of the table in the screen. When this attribute is not specified then the table sets its width according to the width of the row containing the longest data.

`border` Attribute

The **`border`** attribute is used to specify the border width of the table grid. The value as specified corresponds to the number of pixels to construct the grid. If this value is set to '`0`' then there will not be any visible border.

NOTES

`cellspacing` Attribute

The **`cellspacing`** attribute is used to specify the space between the adjacent cells in a table in the form of pixel count.

`cellpadding` Attribute

The **`cellpadding`** attribute is used to specify the width of padding around a table cell in pixels. In other words it introduces a frame of width, specified in this attribute value around the cell. Try the following HTML in your browser:

```
<html>
<head>
<title>HTML table example</title>
</head>
<body>
<h3>HTML table example</h3>
    <table border="0" cellpadding="1">
    <tr>
    <td bgcolor="green"><b>Name</b></td>
    <td bgcolor="green"><b>Age</b></td>
    </tr>
    <tr>
    <td bgcolor="green">Albert Einstien</td>
    <td bgcolor="green">85</td>
    </tr>
    <tr>
    <td bgcolor="green">Issac Newton</td>
    <td bgcolor="green">78</td>
    </tr>
    </table>
</body>
</html>
```

The output on the screen will be displayed as follows:



Captionable Row Element

The <caption> tag is used to add a caption row to the table. Adding a caption tag to the table adds a row at the top of the table which act as a caption for the table describing what the table is about.

Table Row Element

The <tr> tag is used to create a table row. The element <tr> contains one or more <td> or <th> tags.

Table 4.14 shows the attributes associated with the <tr> tag.

Table 4.14 *Attributes of the <tr> Tag*

Attribute	Value	Description
align	right left center justify char	Aligns the content in a table row.
bgcolor	rgb(x,x,x) #xxxxxx colorname	Specifies a background color for a table row.
char	Character	Aligns the content in a table row to a character.
charoff	Number	Sets the number of characters the content will be aligned from the character specified by the char attribute.
valign	top middle bottom baseline	Vertically aligns the content in a table row.
class	Classname	Specifies a classname for an element.
dir	rtl ltr	Specifies the text direction for the content in an element.
id	Id	Specifies a unique id for an element.
lang	language_code	Specifies a language code for the content in an element.
style	style_definition	Specifies an inline style for an element.
title	Text	Specifies extra information about an element.
xml:lang	language_code	Specifies a language code for the content in an element in XHTML documents.

NOTES

Table Cell Element

An HTML table's cells are of two types described as follows:

- Header cell as defined by the <th> element.
- Data cell as defined by the <td> element.

***Note*:** The text in the <th> tag is centered and bold while the text in <td> tag is normal and left aligned.

Table 4.15 shows the attributes associated with the tag <th>.

Table 4.15 *Attributes of the <th> Tag*

Attribute	Value	Description
abbr	Text	Specifies an abbreviated version of the content in a cell.
align	left right center justify char	Aligns the content in a cell.
axis	category_name	Categorizes cells
bgcolor	rgb(x,x,x) #xxxxxx colorname	Specifies the background color for a cell.
Char	character	Aligns the content in a cell to a character.
charoff	number	Sets the number of characters the content will be aligned from the character that specified by the char attribute as.
colspan	number	Specifies the number of columns a cell should span.
height	Pixels, %	Sets the height of a cell.
nowrap	Nowrap	Specifies that the content inside a cell should not wrap.
rowspan	number	Sets the number of rows that a cell should span.
scope	col colgroup row rowgroup	Defines a way to associate the header cells and data cells in a table.
valign	top middle bottom baseline	Vertically aligns the content in a cell.
width	Pixels, %	Specifies the width of a cell.

NOTES

Table 4.16 defines the attributes associated with the `<td>` tag as follows:

Table 4.16 *Attributes of the* `<td>` *Tag*

Attribute	Value	Description
abbr	Text	Specifies an abbreviated version of the content in a cell.
align	left right center justify char	Aligns the content in a cell.
axis	category_name	Categorizes cells.
bgcolor	rgb(x,x,x) #xxxxxx colorname	Specifies a background color for a cell.
char	character	Aligns the content in a cell to a character.
charoff	number	Sets the number of characters the content will be aligned from the character specified by the char attribute.
colspan	number	Specifies the number of columns a cell should span.
headers	headercells'_id	Specifies the table headers related to a cell.
height	pixels %	Sets the height of a cell.
nowrap	Nowrap	Specifies that the content inside a cell should not wrap.
rowspan	number	Sets the number of rows a cell should span.
scope	col colgroup row rowgroup	Defines a way to associate header cells and data cells in a table.
valign	top middle bottom baseline	Vertically aligns the content in a cell.
width	pixels %	Specifies the width of a cell.

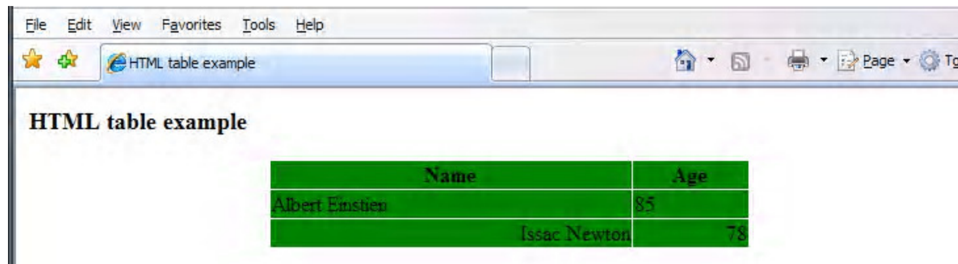
Aligning the Content of Cells and Rows

By default, the content in a cell is aligned to the bottom left. This alignment can be changed by specifying the position in the **`align`** and **`valign`** attribute. If the attribute value is specified in `<tr>` then it is applied to the whole row, but if the attribute is also mentioned in `<td>` then it will overwrite the alignment as specified in `<tr>`. Try the following HTML in your browser:

```
<html>
<head>
<title>HTML table example</title>
</head>
<body>
<h3>HTML table example</h3>
    <table border=”0" cellpadding=”1" cellspacing=”1"
align=”center” width=”50%”>
    <tr align=”center” valign=”center”>
    <td bgcolor=”green”><b>Name</b></td>
    <td bgcolor=”green”><b>Age</b></td>
    </tr>
    <tr>
    <td bgcolor=”green” align=”left”>Albert Einstien</td>
    <td bgcolor=”green” align=”left”>85</td>
    </tr>
    <tr align=”center”>
    <td bgcolor=”green” align=”right”>Issac Newton</td>
    <td bgcolor=”green” align=”right”>78</td>
    </tr>
```

```
    </table>
</body>
</html>
```

The output on the screen will be displayed as follows:



NOTES

In the first row, the alignment was set to center so that the text in all the cells in the first row is centered. In the second row, no alignment is specified at `<tr>` instead the alignment is set to the `<td>` level and accordingly the text is aligned. In the third row, the center alignment is specified but the right alignment specified in `<td>` overwrites the `<tr>` alignment.

Coloring Table with **bgcolor**

The **bgcolor** attribute can be used to specify the background color of the table cell. When specified in the <tr> tag, it applies to the whole row and when specified in the <td> tag, it applies to that particular cell only. If **bgcolor** is specified in both the tags then the color value specified in <td> overwrites the color value in <tr>.

colspan and **rowspan** Attributes

The **colspan** attribute is used to merge the cells into one single row. The number of cells to be merged is specified as the value of **colspan** attribute. The **rowspan** attribute is used to merg cells into a single column. The number of cell merged is specified as value of rowspan attribute. Try the following HTML in your browser:

```
<html>
<head>
<title>HTML table example</title>
</head>
<body>
<h3>HTML table example</h3>
    <table border=”1">
    <tr>
    <td><b>Cell1</b></td>
    <td><b>Cell2</b></td>
    <td><b>Cell3</b></td>
    </tr>
    <tr>
    <td colspan=”2">Cell 1</td>
    <td>Cell2</td>
    </tr>
</table>
</body>
</html>
```

NOTES

The output on the screen will be displayed as follows:



In the second row, specifying the `colspan = '2'` merges the two cells in that row.

Setting Cell Width and Height

The `height` and `width` attributes are used to set the height and width of a cell, respectively. Generally, the `width` parameter is not used as often as compared to the `height` parameter.

Specifying the **`width`** attribute at the table level tells the browser that the table will occupy full window width. Specifying the **`width`** attribute at the column level specifies the column which will take how much width on screen. Generally, the `width` parameter is specified in the first row only and the subsequent rows are formatted in same width.

You can use different height and width parameters for every row. But the browser will format the table using the first row only.

Putting Images and other Elements Inside Table Cells

Unlike text, you can place any content inside a table cell. The content can be an image, data input control, or even a table. Table is used to render the display in a row-column format. This formatting can be used to format any content. Try the following HTML in your browser:

```
<html>
<head>
<title>HTML table example</title>
</head>
<body>
<h3>HTML table example</h3>
    <table border="0" width="100%">
    <tr>
    <td width="20%">TextBox:</td>
    <td><input type="text"/></td>
    </tr>
    <tr>
    <td width="20%">Choice List</td>
    <td><select>
       <option>Option 1</option>
       <option>Option 2</option>
       <option>Option 3</option>
</select></td>
    </tr>
    <tr>
    <td width="20%">Checkbox</td>
    <td><input type="checkbox"/></td>
    </tr>
```

```
    </table>
</body>
</html>
```

The output on the screen will be displayed as follows:



NOTES

Similarly, images can also be inserted into a cell.

***Note*:** Make sure that the image is present with the given names in the same folder wherein your HTML page exists.

Creating Page Margin with Tables

HTML tables can be used to create margins in the display inside the browser window. This can be done by inserting a empty column of width of desired margin. Try the following HTML in your browser:

```
<html>
<head>
    <title>Table Example</title>
<head>
<body>
    <table border="0" width="100%">
       <caption>Currency List</caption>
       <tr >
          <th align="left">Country</th>
          <th align="left">Capital</th>
          <th align="left">Currency</th>
       </tr>
       <tr >
          <td>India</td>
          <td>New Delhi</td>
          <td>Rupee</td>
       </tr>
       <tr >
          <td>United Kingdom</td>
          <td>London</td>
          <td>Pound</td>
       </tr>
       <tr >
          <td>United States of America</td>
          <td>Washington</td>
          <td>Dollars</td>
       </tr>
       <tr >
          <td>Japan</td>
          <td>Tokyo</td>
```

NOTES

```
            <td>Yen</td>
        </tr>
    </table>
</body>
</html>
```

The output on the screen will be displayed as follows:

Currency List

Country	Capital	Currency
India	New Delhi	Rupee
United Kingdom	London	Pound
United States of America	Washington	Dollars
Japan	Tokyo	Yen

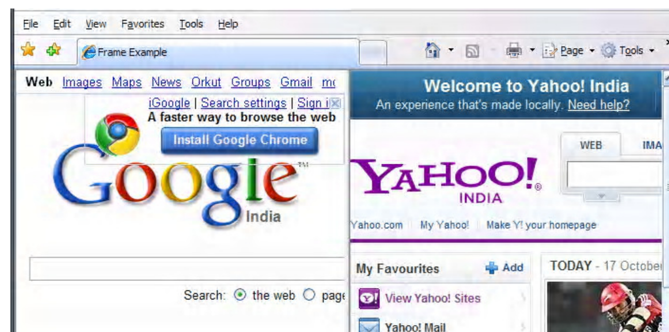
You can see the difference between the display as compared to that in the previous example. By introducing an empty column in the table, a margin is created. Basically, it is not a margin but visually it seems like a margin. An empty column without any text, added in the beginning does the work of creating a margin.

4.10 FRAME

With HTML frames, more than one Web page can be displayed in a single browser window. HTML frames allow an HTML programmer to present the Web pages in different views. Different views offer a programmer a way to keep certain information visible while other views are either replaced or scrolled. Try the following HTML in your browser:

```
<html>
<head>
    <title>Frame Example</title>
</head>
<frameset cols="50%,50%">
    <frame src="http://www.google.com" />
    <frame src="http://www.yahoo.com" />
</frameset>
</html>
```

The output on the screen will be displayed as follows:



The above HTML source creates two frames in the window. It opens google.com in one window and opens yahoo.com in the other. This way, you can view different views in a single window.

NOTES

Creation

An HTML frame is created using the <frameset> tag. In the above example, no <body> tag was present in the HTML source. The <frameset> tag is a replacement of the <body> tag which defines a partition in the browser window. Each partition renders a different Web page containing their body separately.

If you go to any of the frame windows, right click the mouse and select 'View Source', you will find that the source of both frames is different.

Tag <frameset> defines the container which will be subsequently partitioned into the frames. Either the partition can be in rows or in columns. The above example partitions the window into two partitions, assigning 50 per cent area to both the frames.

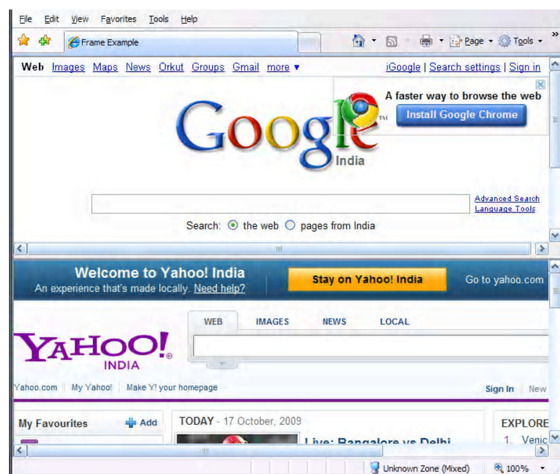
The <frame> tag is used to define an actual frame in the partition. Its **src** attribute points to the Web page that is needed to be opened in this frame.

***Note*:** In the <frameset> tag, either there will be rows attribute or cols (columns) attribute.

Try the following HTML in your browser:

```
<html>
<head>
    <title>Frame Example</title>
</head>
<frameset rows="50%,50%">
    <frame src="http://www.google.com" />
    <frame src="http://www.yahoo.com" />
</frameset>
</html>
```

The output on the screen will be displayed as follows:



Changing the partition attribute from columns to rows in the above example has changed the presentation view of the page. The partition which was initially divided into columns are divided into rows now.

***Note*:** There is no need to specify explicitly the size of the last frame. Instead '*' can be used, which tells to use the remaining width. In the above example, rows= '50%,50%' is same as rows= '50%,*'.

NOTES

Table 4.17 shows the attributes associated with the <frameset>tag.

Table 4.17 *Attributes of the* <frameset> *Tag*

Attribute	Value	Description
cols	pixels % *	Specifies the number and size of columns in a frameset.
rows	pixels % *	Specifies the number and size of rows in a frameset.
class	classname	Specifies a classname for an element.
id	Id	Specifies a unique id for an element.
style	style_definition	Specifies an inline style for an element.
title	Text	Specifies extra information about an element.

Table 4.18 shows the attributes associated with the <frame> tag.

Table 4.18 *Attributes of* <frame> *Tag*

Attribute	Value	Description
frameborder	0 1	Specifies whether or not to display a border around a frame.
longdesc	URL	Specifies a page that contains a long description of the content of a frame.
marginheight	pixels	Specifies the top and bottom margins of a frame.
marginwidth	pixels	Specifies the left and right margins of a frame.
Name	name	Specifies the name of a frame.
noresize	noresize	Specifies that a frame cannot be resized.
scrolling	Yes No auto	Specifies whether or not to display scrollbars in a frame.
Src	URL	Specifies the URL of the document to show in a frame.
Class	classname	Specifies a classname for an element.
Id	Id	Specifies a unique id for an element.
Style	style_definition	Specifies an inline style for an element.
Title	text	Specifies extra information about an element.

Note: If you are using DTD to validate page, then make sure that doctype is set to frameset DTD otherwise your page will be invalidated by the browser.

Also, the <body> tag cannot be used together with the **frameset** element.

4.11 FORM AND STYLE SHEET



FORM: Elements are elements which allow the user to input data on the screen via text boxes, choice lists, check boxes, radio buttons, text areas, etc

HTML forms are used to take different types of user inputs and pass on the data to the server. **Form** elements are elements which allow the user to input data on the screen via text boxes, choice lists, check boxes, radio buttons, text areas, etc. An HTML form is defined using the <form> tag. When the form is submitted, then the data of all the elements as defined inside the form tag is passed on to the server. Try the following HTML in your browser:

```
<html>
<head>
    <title>HTML Form Example</title>
</head>
<body>
    <h3>Registration Form</h3>
    <form name=”registration_form” action=”register.jsp”
method=”post”>
    <table border=”0">
       <tr>
          <td><b>First Name</b>:</td>
          <td width=”20px”></td>
          <td><input type=”text” name=”fname” width=”50" /
></td>
       </tr>
       <tr>
          <td><b>Last Name</b>:</td>
          <td width=”20px”></td>
          <td><input type=”text” name=”lname” width=”50" /
></td>
       </tr>
       <tr>
          <td><b>Sex</b>:</td>
          <td width=”20px”></td>
<td><input type=”radio” name=”sex” value=”M” /
> Male   <input type=”radio” name=”sex”
value=”F” /> Female</td>
       </tr>
       <tr>
          <td><b>Age</b>:</td>
          <td width=”20px”></td>
<td><input type=”text” name=”age” width=”50" /></td>
       </tr>
       <tr>
          <td><b>Address</b>:</td>
          <td width=”20px”></td>
          <td><input type=”text” name=”address” width=”50"
/></td>
       </tr>
       <tr>
          <td colspan=”3" align=”center”><input type=”submit”
value=”Submit”/></td>
       </tr>
    </table>
    </form>
</body>
</html>
```

The output on the screen will be displayed as follows:

NOTES

NOTES

In the above example, a form is defined. All the elements which are defined inside the `<form>` tag are posted to the server. It is very important to assign unique name values to every `<form>` element that we want to post. The server extracts these values using the name of these elements.

The attribute **`action`** defines the server page to which the form values will be posted. In the above example, we have defined a Java Server Page whose name is 'server.jsp' which will be receiving these posted form values.

The attribute **`method`** defines the method used to send the request to the server. Request can be of two types **`get`** or **`post`**. In **`get`** request, there is limitation on the number of parameters that can be posted to the server while in **`post`** request, this is no limitation.

As said before, it is very important to assign a unique name to every element in a form. On the server side, the values which are posted are retrieved using this **`name`** attribute as the key. So, an HTML programmer has to take precaution to assign a unique name to every element inside the form.

Table 4.19 shows the attributes associated with the `<form>` element.

Table 4.19 *Attributes Associated with* `<form>` *Element*

Attribute	Value	Description
action	URL	Specifies where to send the form data when a form is submitted.
accept	MIME_type	Specifies the types of files that can be submitted through a file upload.
accept-charset	Charset	Specifies the character sets that the server can handle for form data.
enctype	application/x-www-form-urlencoded multipart/form-data text/plain	Specifies how form data should be encoded before being sent to a server.
method	get post	Specifies how to send form data.
name	Name	Specifies the name for a form.
target	_blank, _self, _parent, _top, framename	Specifies where to open the action URL.

If you are using Apache Tomcat as your Web server then you can use the following code as your server side page to receive the request:

```
<%
    String fname=request.getParameter("fname");
    String lname=request.getParameter("lname");
    String age=request.getParameter("age");
    String sex=request.getParameter("sex");
    String address=request.getParameter("address");
    out.print("Name = " + fname + " " + lname + "<br/>");
    out.print("Age = " + age + "<br/>");
    out.print("Sex = " + sex + "<br/>");
    out.print("Address = " + address + "<br/>");
%>
```

Save this file as 'register.jsp' and be sure that this file is in the same folder as your HTML page.

When the submit button is clicked, all the data on the form is posted to the server, and the server page 'register.jsp' will receive the data and will process your request.

NOTES

As you can see in the server side code, the request is received and the parameters are read by their name which were mentioned as the name of the elements in the form in your HTML page. So, it is very important to assign unique names to all elements which we want to post on the server side.

***Note*:** Always use the `post` request, as it does not restrict any limit on the number of parameters to be posted. Always assign unique names to elements in the form.

4.11.1 Input Types in Form Element

Here, we will discuss some more types of form elements.

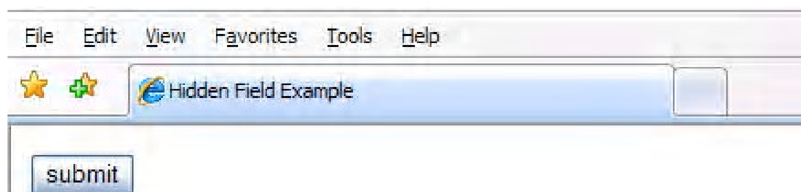
Hidden Field

HTML allows you to have a hidden field. This field has value which is posted to the server side but it is not visible on the page display. Try the following HTML in your browser:

```
<html>
<head>
    <title>Hidden Field Example</title>
</head>
<body>
    <form name="example_form" action="register.jsp"
method="post>
    <input type="hidden" name="fname" value="Albert" />
    <input type="hidden" name="lname" value="Einstien" /
>
    <input type="hidden" name="age" value="85" />
    <input type="hidden" name="sex" value="M" />
    <input type="hidden" name="address" value="PO Box 123
USA" />
    <input type="submit" value="submit" />
       </form>

       </body>
       </html>
```

The output on the screen will be displayed as follows:



You will see only a button on the screen. Rest all the fields are hidden, they are not visible on the page. When you click the submit button, these values are posted to server to, 'register.jsp'. If you are running on Apache Tomcat and 'register.jsp' is present on your server, you will see the values which are posted to the server upon processing this request.

Password Field

The password field is used to mask the input that you enter in a text box. You type the actual text but it is displayed as '*' on the screen. But when it is posted, its actual value is not posted as '*'. The password field is used for entering passwords on the form. Try the following HTML in your browser:

```
<html>
<head>
    <title>Password Field Example</title>
</head>
<body>
    <form  name=”example_form”  action=”register.jsp”
method=”post>
    <input type=”password” name=”fname” />
    <input type=”password” name=”lname” />
    <input type=” password” name=”age” />
    <input type=” password” name=”sex” />
    <input type=” password” name=”address”  />
    <input type=”submit” value=”submit” />
</form>

</body>
</html>
```

The output on the screen will be displayed as follows:

You can see on screen that anything you enter in the text box is masked. But when you click on the submit button, the original text which you enter will be posted to the server not the masked character. If you are running on Apache Tomcat and ‘register.jsp’ is present, then you will see the contents posted to the server when your request is processed.

Submit Button

The submit button is used to submit data to the server. When the submit button is clicked, the data on your form is posted to the server. It should not be confused with a normal button. In case of a normal button, there is no action on clicking of the button.

Cascading Style Sheet

Style sheets are a form of separation of the content of HTML document and its presentation. An HTML file contains its content and structure but does not define its visual layout. Instead, its visual layout is defined in an external style sheet using a style sheet language called CSS (Cascading Style Sheet). This approach is called as ‘Separation of Concern’. Your presentation semantics are separate and your content structural semantics are separate.

CSS is a style sheet language to define the presentation semantics for a document written in HTML. The primary aim of CSS is to separate document content from document presentation. This separation improves content accessibility, and provides control and flexibility over presentation specifications.

NOTES

There are two ways of associating style to an element. One is to mention in the style attribute of the element or the second option is define the style in an external CSS file and link the file to your HTML document.

CSS specifies a priority scheme, so that it can determine which style rule should be applicable if multiple rules match against a particular element.

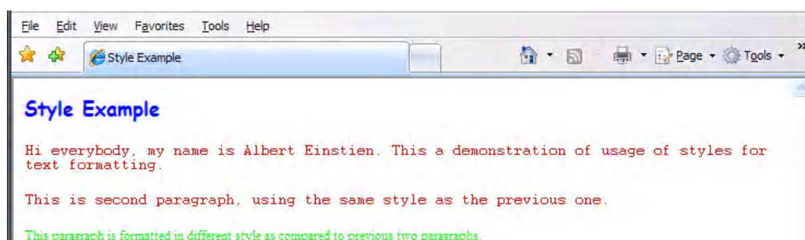
Multi-purpose Internet W3C maintains the specifications for CSS. Mail Extensions or MIME type (Internet Media type) text/css is registered for use with CSS.

Prior to CSS, all the presentational semantics for a Web page were specified only in the HTML markup. Font colors, elements, alignment, borders and sizes had to be explicitly and repeatedly defined in the HTML markup. CSS provides the mechanism to move all the presentational information to a separate document (style sheet) resulting in a simpler HTML markup.

Try the following HTML in your browser:

```
<html>
<head>
    <title>Style Example</title>
</head>
<body>
    <h3 style="color:0000ff;font-family:comic sans ms;">Style
Example</h3>
    <p style="color:ff0000;font-family:courier;font-size:6">
Hi everybody, my name is Albert Einstien. This a demonstration
of the usage of styles for text formatting.
    </p>
    <p style="color:ff0000;font-family:courier;font-size:6">
    This is second paragraph, using the same style as the
previous one.
    </p>
    <p style="color:00ff00;font-family:times new roman;font-
size:12">
    This paragraph is formatted in a different style as
compared the previous two paragraphs.
    </p>
</body>
</html>
```

The output on the screen will be displayed as follows:



In the above example, we defined different styles by setting inline styles for the elements. The following is an example using CSS:

```
.header {
font-family:comic sans ms;
```

NOTES

```
color:#0000ff;
};
.para1{
font-family:courier;
color:#ff0000;
font-size:6;
};
.para2{
font-family:times new roman;
color:#00ff00;
font-size:12;
};
```

Save the above style definition as 'theme.css'. Make sure that this file is in the same folder as your HTML file. Try the following HTML in your browser:

```
<html>
<head>
    <title>Style Example</title>
<link rel="stylesheet" type="text/css" href="theme.css" />
</head>
<body>
    <h3 class="header">Style Example</h3>
    <p class="para1">
  Hi everybody, my name is Albert Einstien. This
is a demonstration of the usage of styles for text
formatting.
    </p>
    <p class="para1">
    This is second paragraph, using the same style as the
previous one.
    </p>
    <p class="para2">
    This paragraph is formatted in different styles
as compared to the previous two paragraphs.
    </p>
</body>
</html>
```

The output of the above HTML will be the same as in the previous example. But in this example, we have separated the presentation semantics and the content semantics. If there is a need to change the presentation semantics, only the CSS file needs to be changed and the change is then reflected on the screen.

Properties of CSS

Properties of CSS are divided into several groups according to their behaviour. These groups are as follows:

- Background
- Border and outline
- Dimension
- Font
- Margin

Check Your Progress

19. Why `<table>` tag is used?
20. Why `align` attribute is used?
21. Why `<cellpadding>` attribute is used?
22. Why `<caption>` tag is used?
23. What does HTML frames allow?
24. What is the use of `<frameset>` tag?
25. Why HTML forms are used?

- Padding
- Positioning
- Text
- Pseudo classes

NOTES

4.12 UNDERSTANDING XHTML

XHTML means 'eXtensible HyperText Markup Language'. It can be interpreted as 'the stricter and cleaner version of HTML'. It is a W3C recommended XML application and is almost identical to HTML 4.01. You can understand it as the combination of HTML and XML which consist of elements of HTML 4.01 defined by the syntax of XML.

Many Websites are coded in bad HTML. An interesting thing about HTML is that its code continues to work well even if it does not strictly follow the HTML rules.

```
<html>
<head>
<title>You are working in XHTML</title>
<body>
<h1>It is possible to work with HTML even when you do not
follow its rules strictly.
<h2> You just have a look at the code of this document. You
have not closed the html tag. But it is still going to work.
This can be interpreted as smartness of the browser which
interprets incomplete html.
</body>
```

With the advancement in technology, many browsers are available in market today. Some browsers run the Internet on computers, whereas some run on mobile phones or other small devices. Some browsers, like the ones working on mobile phones, may not be able to interpret incomplete HTML, or in other words, bad HTML. By combining HTML and XML, a markup language (XHTML) has been developed which can work with all the browsers.

4.12.1 Components of an XHTML Document

An XHTML document can be divided into the following components:

- DOCTYPE declaration
- <head> section
- <body> section

Check out the following code:

```
<!DOCTYPE ...>
<html>
<head>
<title>... </title>
</head>
<body> ... </body>
</html>
```

NOTES

This code illustrates the basic document structure of an XHTML document.

Note: The DOCTYPE declaration is always written as a first line in an XHTML document!

An example of XHTML is as follows:

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Learning XHTML</title>
</head>
<body>
<b>This is your first XML document</b>
</body>
</html>
```

4.12.2 Characteristics of XHTML

Important characteristics of XHTML are as follows:

(i) XHTML Elements Need to be Always Closed: As already discussed, in HTML an incomplete code may work. However, when making a page according to XHTML standard, it is important to check that all XHTML elements are closed.

(ii) Every 'Empty Element' Must be Closed: HTML tags can broadly be divided in the following two categories:

- **Paired tag:** Paired tags are the tags which need to be closed. For example:
  a. <html>...</html>
  b. <body>....</body>
  c. <style>....</style>
  d. <p> .....</p>
- **Singular Tag (also called Empty Tag):** Singular tags are the ones which do not need to be closed. For example:
  e.
  f. <hr>
  g. <img>

According to the XHTML rules, empty elements must either have an end tag or the start tag must end with />.

(iii) In XHTML Elements Must be Properly Nested.

(iv) XHTML Elements Must be in Lower Case: HTML is notcase sensitive, i.e., <html>, <Html>, <HTML> all will work. However, according to the XHTML specification, the tag names and attributes need to be lower case.

(v) XHTML Document Needs to have One Root Element: In case of XHTML, all elements must be nested within the <html> root element. All other elements in XHTML are sub-elements or children of the root element. Other elements can further have sub-elements. You need to be careful as sub-elements need to be paired. Further, child elements need to be correctly nested within their parent element.

NOTES

Defining XHTML Attributes

- Write attribute names always in lower case. Check out the following code

  Incorrect XHTML
  ```
  <table HEIGHT="40 %">
  ```
  Correct XHTML
  ```
  <table height="40 %">
  ```
  Here `height` is an attribute of table and needs to be written in lower case.

- Always write attribute values in quotes. Check out the following code:

  Incorrect XHTML
  ```
  <table height=40%>
  ```
  Correct XHTML
  ```
  <table height="40 %">
  ```
  Here you are assigning value to attribute `height`, thus the need to put it in quotes.

- You cannot minimize attributes.

  Incorrect XHTML
  ```
  <input checked>
  <input readonly>
  <input disabled>
  <option selected>
  <frame noresize>
  ```
  Correct XHTML
  ```
  <input checked="checked" />
  <input readonly="readonly" />
  <input disabled="disabled" />
  <option selected="selected" />
  <frame noresize="noresize" />
  ```

Here is a list of the minimized attributes in HTML and how they should be written in XHTML as shown in Table 4.20:

Table 4.20 *Attributes in HTML and their Conversion into XHTML*

HTML	*XHTML*
compact	compact="compact"
checked	checked="checked"
nohref	nohref="nohref"
noshade	noshade="noshade"
Nowrap	nowrap="nowrap"
multiple	multiple="multiple"
noresize	noresize="noresize"

- Use XHTML DTD to define mandatory elements:

Points to Remember about XHTML Elements

- All XHTML documents need to have a `DOCTYPE` declaration.

NOTES

- The `html`, `head` and `body` elements must be present.
- Title must be present inside the `head` element.

Check out a minimum template for an XHTML document:

```
<!DOCTYPE......................>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Title goes here</title>
</head>
<body>
</body>
</html>
```

Note: As discussed before, `DOCTYPE` declaration is not a part of the XHTML document itself. As in XML `DOCTYPE` declaration is not an XHTML element and so no need to have a closing tag.

4.12.3 Steps to Convert HTML to XHTML

1. Add DOCTYPE Definition

Check out the following code. `DOCTYPE` declaration has to be added as the first line of an XHTML document:

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

In case you are using the `Strict` or Frameset DTD, you need to use the appropriate declaration.

Note: You need to have a `DOCTYPE` declaration on the Web page for validating the XHTML.

2. Change Lowercase Tags and Attribute Names to Uppercase

XHTML is case-sensitive and only accepts the lowercase tags and attributes. You need to 'find-and-replace' all uppercases with lowercases if they are found in tags and attributes.

3. Quote all Attribute Values

According to the XHTML 1.0 recommendation, all the attribute values in the XHTML document must be quoted.

4. Replace Empty Tags

You need to close every open tag. Empty tags, such as `<hr>`, `<br>` and `<img>`, used in HTML are not allowed in XHTML. You need to replace `<hr>` and `<br>` tags with `<hr />` and `<br />`.

Note: There are many online tools available that you can use for validating XHTML. The official tool is available at the W3C Website called W3C DTD Validator. There are other free tools available, such as Dave Raggett's HTML TIDY which is a free tool and can be used for cleaning up HTML codes. Such tools help to identify where attention is required to make pages that attract people or make pages that are accessible to people with disabilities.

4.13 DHTML

NOTES

DHTML stands for Dynamic Hyper Text Markup Language or simply Dynamic HTML. It is a combination of HTML, CSS, JavaScript, and Document Object Model (DOM). DHTML can be described as the art of making interactive and dynamic HTML documents.

DHTML allows the scripting languages to change the variables and attribute values of the element on screen, which in turns affects the look and feel of an HTML page. Otherwise, an HTML page is static after it is fully loaded into the browser window.

If we talk about a dynamic wee page then it has a much wider perspective and broader concept. This includes the pages that are created by client-side scripting such as JavaScript or by server-side scripting like Active Server Pages (ASP), or Java Server Pages (JSP), or PHP, etc.

Here, our focus will be only on the Dynamic HTML pages created by client-side scripting. Discussing Server side scripting is out of scope of this book.

The major disadvantage of DHTML is that it is difficult to debug and develop Web pages due to the difference in support as extended by Web browsers for scripting languages. A Web page may work in Internet Explorer but may not work in Safari.

Object Structure

DOM (Document Object Model) is an independent platform and a language neutral interface that allows the scripts and programs to access/update the document content, its structure, and style dynamically. The document can be processed after the display and the result of processing can be re-displayed on the same page.

Dynamic HTML is a combination of HTML, CSS, Javascript, allows the document to be dynamic. It provides you with an object-oriented model, in which, every element is an object and is a part of the main document object which represents the Web page, and the document object is part of the window object which represents the browser window.

The DOM defines the objects and properties for all the document elements and the methods to access them. The HTML DOM defines a standard way for accessing and manipulating HTML documents. The DOM presents an HTML document as a tree structure as shown in Figure 4.1:

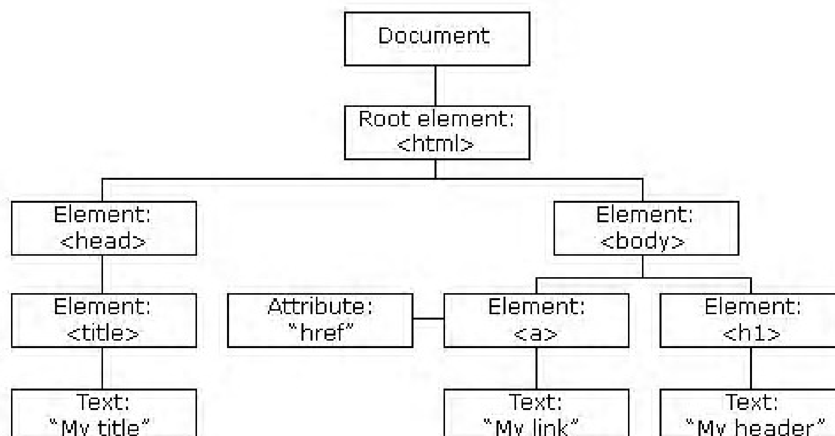


Fig. 4.1 The Tree Structure of an HTML Document

NOTES

The HTML DOM views an HTML document as a tree-structure. The tree structure is known as a node-tree. According to DOM, everything in an HTML document is a node, even the text and comments are nodes, termed as 'Text Nodes' and 'Comment Nodes' respectively. The root node is `<html>`, and all other nodes are contained in this node. It has two child nodes `<head>` and `<body>`. All nodes can be accessed through the tree. Their contents can be modified or deleted, and new elements can be created.

The nodes in the node tree have a hierarchical relationship to each other. The terms parent, child, and sibling are used to describe the relationships. Parent nodes have children. Children on the same level are called siblings (brothers or sisters).

- In a node tree, the top node is called the root.
- Every node, except the root, has exactly one parent node.
- A node can have any number of children.
- A leaf is a node with no children.
- Siblings are nodes with the same parent.

Figure 4.2 illustrates a part of the node tree and the relationship between the nodes as follows:

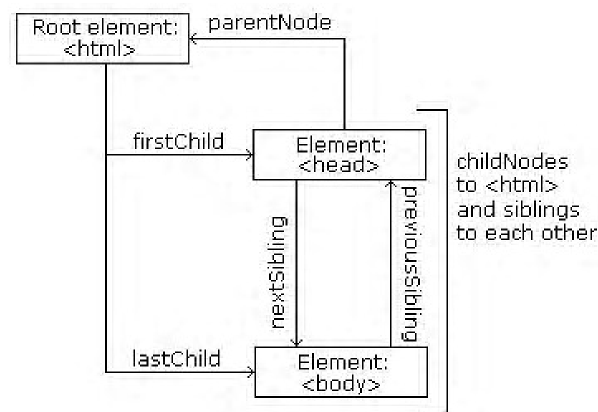


Fig. 4.2 *Relationship between the Nodes*

DOM Properties and Methods: Properties are often referred to as something that is the value of a node. Methods are often referred to as something that is done to add a node. Following are the properties of nodes:

- **x.innerHTML:-** The text value of x.
- **x.nodeName:-** The name of x.
- **x.nodeValue:-** The value of x.
- **x.parentNode:-** The parent node of x.
- **x.childNodes:-** The child nodes of x.
- **x.attributes:-** The attribute nodes of x.

Note: In the above list, x is a node object (an HTML element).

Following are the methods associated with nodes:

- **`x.getElementById(`*`id`*`):`** Get the element with a specified id.
- **`x.getElementsByTagName(`*`name`*`):`** Get all elements with a specified tag name.
- **`x.appendChild(`*`node`*`):`** Insert a child node to x.
- **`x.removeChild(`*`node`*`):`** Remove a child node from x.

Note: In the above list, x is a node object (HTML element).

Window Object: It represents the browser window in which the HTML document is rendered. If a document contains frames or iframes then the browser creates one window object for a document and additional window objects for each frame respectively.

NOTES

The properties associated with the window object have been described in Table 4.21:

Table 4.21 *Properties of the Window Object*

Property	Description
closed	Returns whether or not a window has been closed.
defaultStatus	Sets or returns the default text in the status bar of the window.
document	Returns the document Object.
history	Returns History Object.
length	Sets or returns the number of frames in the window.
location	Returns the location of object.
name	Sets or returns the name of the window.
opener	Returns a reference to the window that created the window.
outerHeight	Sets or returns the outer height of a window.
outerWidth	Sets or returns the outer width of a window.
pageXOffset	Sets or returns the X position of the current page in relation to the upper left corner of a window's display area.
pageYOffset	Sets or returns the Y position of the current page in relation to the upper left corner of a window's display area.
parent	Returns the parent window.
personalbar	Sets whether or not the browser's personal bar (or directories bar) should be visible.
scrollbars	Sets whether or not the scrollbars should be visible.
self	Returns a reference to the current window.
status	Sets the text in the status bar of a window.
statusbar	Sets whether or not the browser's status bar should be visible.
toolbar	Sets whether or not the browser's tool bar is visible (it can only be set before the window is opened and you must have the UniversalBrowserWrite privilege).
top	Returns the topmost ancestor window.

Table 4.22 describes the methods associated with the window object as follows:

Table 4.22 *Methods Associated with the Window Object*

Method	Description
alert()	Displays an alert box with a message and an OK button.
blur()	Removes focus from the current window.
clearInterval()	Cancels a timeout set with **setInterval**().
clearTimeout()	Cancels a timeout set with **setTimeout**().
close()	Closes the current window.
confirm()	Displays a dialogue box with a message and an OK and a Cancel button.
createPopup()	Creates a pop-up window.
focus()	Sets focus to the current window.
moveBy()	Moves a window relative to its current position
moveTo()	Moves a window to the specified position.
open()	Opens a new browser window.
print()	Prints the contents of the current window.
prompt()	Displays a dialogue box that prompts the user for input.
resizeBy()	Resizes a window by the specified pixels.
resizeTo()	Resizes a window to the specified width and height.
scrollBy()	Scrolls the content by the specified number of pixels.
scrollTo()	Scrolls the content to the specified coordinates.
setInterval()	Evaluates an expression at specified intervals.
setTimeout()	Evaluates an expression after a specified number of milliseconds.

4.14 MICROSOFT INTERNET EXPLORER

NOTES

Internet Explorer is a series of graphical Web browsers, developed by Microsoft and included as a part of the Microsoft Windows operating systems. Each of the browser available these days consists of three parts, a controller, client protocol and interpreters, and so the Internet Explorer. The controller receives input from input devices, such as keyboard, mouse, etc. It uses client programs to access the Web pages. After the document has been accessed, the controller uses one of the interpreters to display the document on screen.

4.14.1 Anatomy of a Web Browser: The Toolbar

The various toolbars that are structured in browsers make up the anatomy of a Web browser.



The back button takes you the previous page, the forward button returns you the page from where you went back, home button takes you the home page of the specified Website, stop button stops the browser to load the current page and the print button lets you print a copy of the Web page. The toolbars are available in the Web browser is described as follows:

Back Button: This is shown by an arrow that points to left. Clicking on it brings you back to the previous Web page you had visited.

Forward Button: This is shown by an arrow pointing to the right. Clicking on it returns you to the page from where you just came.

Home Button: This button is recognized by the icon of home in Mozilla and the Internet explorer and clicking on it takes you to the home page you have chosen. If no home page is selected by you then it would bring you to a default home page that is normally a Website of Google, Microsoft or Netscape.

Reload or Refresh Button: This loads the Web page once again. This is normally done when all elements of a Web page are not loaded in the first attempt and file transfer gets interrupted. Also, while downloading a Web page, data is temporarily stored in computer memory and it is told that '**data is cached'.** On accessing this Web page next time, the Web browser accesses it from cache instead of requesting the Web server for retrieving this file. There are certain Web pages that are updated frequently, as with scores of sport events, financial data or news. In such cases you do not get current information, but by reloading it, data gets updated from the Web server.

Print Button: Clicking on it enables you to have a hard copy of current page that is loaded in the browser.

Stop Button: This button is used to stop the browser from loading the current page.

Search Button: This button provides connection to search tools and directories of the Websites of Google, Microsoft or Netscape.

Bookmarks or Favorites Button: Internet explorer shows Favorites button whereas Mozilla and Netscape shows Bookmarks. This enables you to record addresses of Websites for revisiting. On adding a URL in your list, you may come back to that Web page by just clicking the link in the list that you have made and you need not retype entire address.

NOTES

Address Bar: This is situated either next to the toolbar or above it. Where user can type URL or address of the Website to visit. After entering the address here, hit the **Enter** or **Return** key for accessing the site. Alternatively make a click on 'Go' or an Arrow button situated towards the right of address box. By clicking on the small downward triangle towards the right of Location box, a drop-down list appears showing most recent Websites visited by you. This helps you in revisiting a site and you can select the address instead of typing it. Figure 4.3 shows the various buttons in IC or Internet Explorer browser.

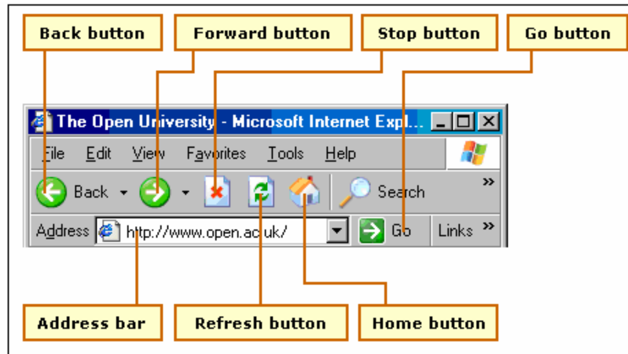
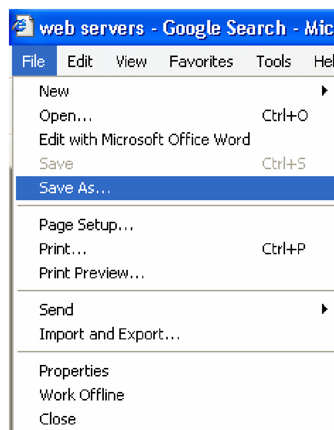


Fig. 4.3 Various Buttons in IE Browser

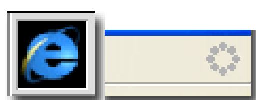
The Menu Bar: This is located at the top of browser window. This enables you to select things with a Web page. You may save the Web page on your hard disk or may like to increase the text size on the page. Most of these choices are same as buttons on the toolbar. Click once on a menu item for accessing the drop-down menu, then select and click to perform action as desired. For example, clicking on File menu shows a dropdown menu from which Save As option appears. You may save the Web page on your hard disk by choosing a location with default file name given by the browser or may choose a name as desired by you.



File menu is used to save the Web page, import or export the files too.

4.14.2 The Access Indicator

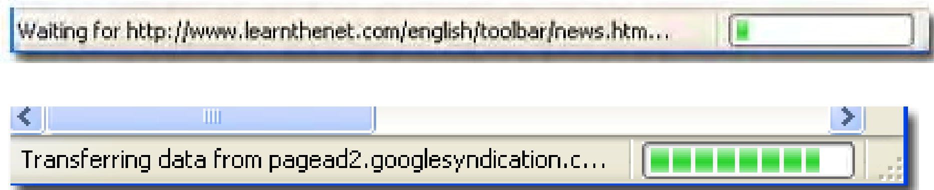
Internet Explorer or Mozilla Firefox, contain small graphics for indicating the activity of the browser. Animation of this image indicates that the browser software as a **client** is attempting to access data from a **server** that is a remote computer.



NOTES

Access Indicator: A server may be located at any remote location, maybe a city in your country or may be in another country or even another continent. The browser can download files from a remote computer to your computer and can also display these on your computer screen. Time taken by this process depends on many factors; speed of net connection, file size, load on the server and traffic on the Internet.

The Status Bar: This is situated at the bottom of the Web browser. Once you try to access a Website this bar shows the progress on transactions in the Web page. For example, when you type the address of the site you are trying to visit, status bar shows whether the Website has been found along with number as well as size of files that are being downloaded.



The Scroll Bar: There is a vertical or horizontal bar located towards the bottom side and right side of the browser respectively. Using this you can scroll a Web page down and up. To do this, place your pointer on arrows, up or down by using left key of the mouse. You may drag the slider by placing the pointer on slider control and holding down the left key of the mouse. There is a scrolling wheel is some mouse that provides an alternative way to navigate a long Web page. In case width of a Web page is more and can not fit the computer screen, you will find a horizontal scroll bar just above status bar. This can be used to move Web page left or right.



4.15 SHOCKWAVE AND LINGO

Shockwave is not a development tool but it is a runtime routine. There are three different versions of Shockwave. Two versions are runtime routines for multimedia applications and the third version is for a graphics package.

Shockwave is a browser plug-in that enables the specific applications, such as Macromedia Inc., Macromedia Director, Macromedia Authorware and Macromedia Freehand to play on an Internet Web site or on an intranet. Although Shockwave is not a development tool, it does enable the applications that have been shocked to include Web-related commands, such as the loading of HTML pages and to become interactive tools for the World Wide Web.

Specifically what are the three packages that can be shocked? Director is a multimedia authoring package, Authorware is a CBT authoring package and Freehand is a vector-based drawing package. Shockwave is a new technology and is fast improving

and expanding. Each week Macromedia posts news on updates and changes to its Shockwave technology, including new Macromedia products capable of being shocked and new browsers capable of playing shocked applications.

NOTES

Shockwave is an Internet browser plug-in that was developed by Macromedia to enable users to view and interact with Web-based multimedia files. Shockwave currently supports the platforms Director, Authorware and Freehand.

Director

Director is a powerful multimedia authoring tool. It has two components, a design package capable of creating sophisticated animations and an extremely powerful programming language, called Lingo. Lingo contains the core of Director's power and adds complex interactive and decision-making components to an already robust design tool.

To add a shocked movie to an HTML document, you use an HTML EMBED tag which comprises several components.

```
<EMBED SRC>
```

The tag begins with <EMBED SRC="??????".

Here SRC= refers to the source or filename of the shocked file to appear (that is, to be embedded) on the Web page.

PALETTE

Director uses the PALETTE= tag, followed by either foreground or background, for example PALETTE=foreground. Foreground causes Director to load the specified palette. The palette is then used for the whole Web page. Background causes Director to ignore its internal palette(s) and to use the computer's system palette. If you do not include the EMBED tag, Director uses PALETTE=foreground as the default. If you do include the EMBED tag, PALETTE=background is also the default. As with all HTML tags, the EMBED tag ends with >.

If your browser does not support shocked files, you will see a broken icon on your Web page. You can avoid broken icons simply by using JavaScript. For this knowledge of Java program is not required, you have to follow a simple template. JavaScript is a programming language that is built into Netscape Navigator 2.0. The key to JavaScript is HTML comment tags, <!–> and <–>. All other browsers ignore the comment tags except Netscape 2.0. When Netscape 2.0 sees a comment tag that is prefaced by <SCRIPT LANGUAGE="JavaScript">, it will read the information inside the comment tags. The format for Shockwave files is as follows:

```
<SCRIPT LANGUAGE="JavaScript">
<!– Information inside this comment area is only recognized by Navigator 2.0.
document.write( '<EMBED SRC="shockedfile" WIDTH=x HEIGHT=y>' );
<!– This ends the hidden JavaScript from non-Navigator 2.0 Browsers –>
</SCRIPT>
```

Because JavaScript is hidden with a comment section, only browsers capable of reading JavaScript will see and display the shocked image. Developers can create multimedia applications using just the design and animation portion of the Director package.

NOTES

Most developers use Lingo to add control and interactivity to Director movies format. Lingo, as Director's programming language, adds the capability of clicking buttons, branching choices, controlling animations, and much more.

Even though Lingo has over 200 commands, numerous additional Web-related commands and extensions have been added to the Lingo language for creating shocked Websites. These extensions can be divided into asynchronous operations and functions.

getNetText uri

`getNetText uri` is the Lingo command that retrieves an HTTP text item for Lingo to read.

Although Lingo is not case sensitive, but protocols specify that all Lingo commands must be written in lowercase and beginning all other words within a group command phrase in uppercase, for example, `startAnimation` or `mouseDown`.

The `uri` portion of the command refers to the Universal Resource Identifier (URI). Universal Resource Locators (URLs) are a subcategory of URIs, as are file extensions. An example of the `getNetText uri` command is:

```
getNetText "http://myWebsite.com/newsletter.text"
```

The `NetDone` command determines if the `getNetText` command has finished. The uri defines the location and name of the specified text.

gotoNetPage uri, target

`gotoNetPage uri` is a Net-related Lingo extension that opens a new URI. It can be another MIME type, such as an HTML page. The target parameter is optional. It references a target frame if you have designed a page using the Netscape frame feature.

preloadNetThing uri

The `preloadNetThing uri` Lingo command preloads an HTTP item into Netscape's disk cache so that you can access or download the item without any delays. An example of a `preloadNetThing` command is:

```
preloadNetThing "http://myWebsite.com/graphics/
logo.gif"
```

Functions

Functions return a result based upon the preceding asynchronous commands. Macromedia added to Lingo a small group of functions that are specific to the Internet. They are referred to as `net` functions and are identified by the inclusion of the letters `net` in their name.

netDone()

If the asynchronous operation is complete, `netDone()` returns a result of `TRUE`. `netDone()` returns a result of `FALSE` if the operation has been started, but has not finished. The `netDone()` command defaults to `TRUE`. An example of the `netDone()` function is as follows:

```
If netDone() then puppetSound "Trumpet"
```

netError()

`netError()` returns one of following three results:

- If an operation has not finished, `netError()` returns an empty (`netError()=EMPTY`) string.

- If an operation has completed successfully, `netError()` returns OK (`netError() = "OK"`).
- If an operation has not completed successfully, `netError()` returns a string that describes the error.

NOTES

netTextResult()

`netTextResult()` returns the result of the `getNetText` command. An example of this command is as follows:

```
If netDone() then
        put netTextResult() into field "Headline"
end if
```

netMime()

`netMime()` returns the MIME type of an HTTP item. Only Director 5 shocked files support this function.

netLastModDate()

`netLastModDate()` returns the last modified date string located in the HTTP header for the item referenced. An example of this function is as follows:

```
put netLastModDate() into field "Page Updated On"
```

Only Director 5 shocked files support this command.

getLatestNetID()

`getLatestNetID()` returns a unique identifier describing the last asynchronous operation started. This function is handy when using the `netAbort` command or the `netDone()` function.

XObjects, XCMDs, XFCNs

`XObjects`, `XCMDs` and `XFCNs` refer to external objects, commands and functions. Although both Windows and Macs can use `XObjects`, only the Mac can use `XCMDs` and `XFCNs`. `XObjects` enable third-party developers to create new features and enhancements for Director 4.0*x*.

`XObjects`, `XCMDs`, and `XFCNs` can be either internal (included within a Director movie) or external (referenced or pointed to by a Director movie). Shocked movies support only external `XObjects`, `XCMDs` and `XFCNs`. In order for a movie to use external `Xobjects`, `XCMDs` or `XFCNs`, these files must be located in the user's Plug-In directory.

Authorware

Authorware is Macromedia's icon-based authoring package. Because of its inherent features, it is primarily used to develop CBT. Like Director, Authorware supports very sophisticated interactivity, including the use of clickable buttons or graphics (hotspots). Currently, Shockwave works only with Authorware 3.5 and on Netscape 2.0 or 2.01.

Although Authorware shocked applications can be used either on the Internet or on intranets, Authorware is currently targeted primarily for use with intranets. Authorware uses the Afterburner tool to convert Authorware applications into shocked applications. Afterburner is available both for Windows and Mac. Afterburner performs the following three functions:

1. It compresses Authorware files and libraries.
2. It divides the files into several segments.

NOTES

3. It creates one map file for locating each segment when needed. Map files are explained later.

The file segments, the map and the flattened external files are assigned MIME types.

application/x-authorware-map	MIME type for the Authorware map.
application/x-authorware-seg	MIME type for a file segment.
application/x-authorware-bin	MIME type for a flattened external file.

Functions and Variables

In Authorware the functions and variables addresses are specific to Internet and intranet environments which focus primarily on references to HTML documents (Net pages) and URLs.

`GoToNetPage(URL), window`

Attaching the `GoToNetPage(URL)` function to a button or hotspot causes Netscape to load the Web page listed in `(URL)`. You can also add a second optional argument to this function. This argument is a target to HTML windows. Windows are a part of Netscape's Frame technology, and you should study Netscape Frame Scripting to understand the use of frames and HTML windows. Table 4.23 lists possible target options for the second (the window) argument.

***Table 4.23** Options for the GoToNetPage Window Argument*

Option	*Meaning*
`_blank`	Causes the Authorware application to remain in a window while a second window displays the new Web page. You can use this method to return to the Authorware application after viewing the Web page.
`_self`	Causes the new Web page to open in the same window as the current page.
`_parent`	Causes the new Web page to open in the same frameset parent window as the current page. When you are using target windows, this option overrides a globally assigned `base` target (the default window). If there is no parent, this argument functions the same as the `_self` argument.
`_top`	Causes the new Web page to open in the body of the current window. This option is equivalent to the `_self` argument. It is also used to return from frame nesting. If you do not include a second argument, the Authorware application closes as soon as it locates the new Web page.

Use a comma to separate arguments; for example, `URL,"_blank"`. An example of the `GoToNetPage(URL)` function is:

```
GoToNetPage("http://www.mysite.com", "_blank")
```

NOTES

NetDownload(URL)

The `NetDownload(URL)` function downloads the file listed in `(URL)` to the user's computer. The full pathname is returned if the operation was successful. This function, in combination with the `onDemand` flag in the map file, enables files to be downloaded only when requested.

`NetDownload(URL)` looks at the map file and downloads to the location listed by the `put` line of the map file that precedes the line entry for the file being downloaded. If the map file does not have an entry for the file being downloaded, it will be downloaded to the `DOWNLOAD` subfolder of the folder containing the Authorware plug-in.

NetPreload(IcondID@"IconTitle")

`NetPreload(IcondID@"IconTitle")` is an asynchronous command that downloads whatever segment files the icon specified by `IconTitle` requires to execute. If the `IconTitle` is a map, interaction, decision, or framework icon, all icons that are either attached to (external) or within (internal) the `IconTitle` are preloaded.

The primary function of the `NetPreload` function is to initiate background preloading of segments required for sounds, movies or graphics, so that the segments are on the user's system when needed.

NetConnected

NetConnected is a flag variable (`1=TRUE`, `0=FALSE`). It tells the Authorware application if it is running via Shockwave. The value is `1` if the application is running under Shockwave and `0` if it is running either under Authorware's runtime routine or in the authoring environment. This variable is useful with calculation icons or decision icons to branch according to the environment on which the Authorware application is playing.

NetLocation

The `NetLocation` flag returns the URL of the current Shockwave application. A useful routine is to use the `NetConnected` flag to determine if the Authorware application is running on the Internet or as a stand-alone application. Then, either retrieve a file off the HTTP server in a file relative to the `NetLocation` if the application is running on the Internet or retrieve the file off the hard drive or a CD-ROM using a local address if the application is running in stand-alone mode.

Freehand

The absolute Macromedia application that a Shockwave plug-in has ported to the Internet is Freehand. Freehand is a cross-platform, vector-based drawing package that integrates special effects via `Xtras` plug-ins, such as Smudge, Fisheye Lens and 3D Rotation, along with powerful page layout potentials to programs, such as PageMaker and QuarkXPress. You can use Freehand to create exciting page-based documents. With the inclusion of Shockwave, these pages can appear on the Web. When a graphic is created in Freehand, you can use Freehand's Page inspector to determine the page size.

LINGO

Lingo is a scripting language developed by John H. Thompson for Adobe Director (formerly Macromedia Director) specifically used for regular desktop applications, interactive kiosks, CD-ROMs and Adobe Shockwave Internet Websites.

Syntax

When Lingo was created, a wordy or verbose syntax was designed to write whole HyperTalk like sentences such as:

NOTES

```
if sprite 5 is visible then go to the frame
```

In the above syntax, with `go to` statement `to` is optional. Lingo was also initially very robust by providing object generation through a notion called `factory`, which led to the language's extensibility through External Factories `XFactories` or `XObjects`.

Although it is still possible to use the verbose syntax, the current version of the language fully supports dot syntax, so that the code looks more like standard programming languages, such as JavaScript or C++. The equivalent in new scripting style is as follows:

```
if sprite(5).visible then _movie.go(_movie.frame)
```

This format uses elements of the Director Object Model (DOM), introduced in Director MX 2004.

Lingo is embedded into Adobe Director. Lingo is an Object Oriented Programming (OOP) language. The following are three types of scripts in Lingo:

- **Behavior** scripts are attached to a sprite or inserted into a frame. Sprite behaviors are often used to give control of the sprite's properties and movement. Frame behaviors can be used to create a pause or delay within a certain frame in the score. Behaviors make it easy to program in an object oriented way. They can also control or interact with other sprites, making them a true object.
- **Movie** scripts can neither be attached to sprites nor can they be instantiated as objects. They are available throughout the program (movie) and are especially useful for holding global handlers and initializing global variables at the start or end of the movie.
- **Parent** scripts are used to create instances of (birth) an object into a variable using the `new` command. These objects can control sprites and other media remotely without being attached to any one sprite. It can be used to control data or other non-displayed items and are useful for recursion routines, such as path finding. A parent script can be used to create or destroy an object at anytime, freeing them from the confines of the score that a behavior is limited to.

Behavior and **parent** scripts support object oriented programming paradigm. **Movie** scripts are not OOP oriented, however they can still be used to make black-box handlers where other objects can input raw data and receive answers back without knowing the inner workings of the box.

Lingo supports object **inheritance** by a slightly idiosyncratic system such that a script can have an **`ancestor`** property which references another object usually also a script although other objects, such as cast members can also be ancestors. Properties and methods of the ancestor are inherited by the parent. Behavior scripts are also a kind of ancestor of the sprites to which they are attached, since properties and methods of the behavior can be accessed by reference to the sprite itself. Thus it refers a kind of multiple inheritance as one sprite may have several behaviors.

`XObjects`

Lingo was also extensible through External Factories `XFactories` or `XObjects` later called Lingo Xtras, which provided programmatic extensions to Director, for example, controlling external media devices, such as CD-ROM and Video tape players through Macintosh serial port. `XObject API` was openly available to developers and media device producers, which added to the popularity and versatility of Lingo. Macromind was very active in positioning the `XObject API` as standard for external media devices. Lingo was quickly adopted by growing multimedia community and the already popular Micromind Director product.

4.16 NETSCAPE EXTENSIONS

Netscape extensions focus primarily on making documents appear enhanced. They are an excellent method to improve the appearance and impact of a home page in an increasingly Netscape dominated Web. Netscape extensions should be avoided when publishing reports, articles and other items where the textual content is more important. They could cause problems when HTML 3 is finalized. The following are some important features of Netscape extensions:

- New tags include NOBR, WBR, FONT, BASEFONT, CENTER and BLINK.
- Many other tags have new attributes to provide more control over the appearance of text and graphics.
- Tables are supported.
- Documents can include tiled image backgrounds and you can change the color of text on a document.

Netscape extensions are unauthorized additions to the HTML 2.0 standard. Originally the Netscape extensions only worked with the Netscape Navigator browser. However, Mosaic 3.0+ browsers will support many of the Netscape extensions. If you use Netscape extensions in your document you should indicate this at the top of your document. Even though unauthorized, the Netscape extensions have become commonly excepted tags and are used in many Web documents.

<CENTER>...</CENTER>

This is the center tags and is one of the most popular Netscape extensions. The `<CENTER>` tag allows you to center text and graphics on a page. The banner image and divider bars on this page are all centered using the `<CENTER>` tag. `<HR>` tag attributes Netscape allows you to manipulate the size and width of the `LINE RULE` tag. The `SIZE` attribute indicates the thickness of the rule line (the default is 2). The `WIDTH` attribute with `ALIGN` allows you to indicate the horizontal width of the rule line, either in pixels or as a percent of the page. The display resolution of the screen will affect the display.

o <HR ALIGN=CENTER WIDTH=80%

o <HR ALIGN=CENTER WIDTH=40%

o <HR ALIGN=CENTER WIDTH=20%

o <HR ALIGN=CENTER WIDTH=10%

Unordered List Tag Extensions: The following are the unordered list tag extensions:

- `TYPE=DISC`: This is used for a solid bullet, the default is `</LI>`.
- `TYPE=CIRCLE`: This is used for a hollow bullet.
- `TYPE=SQUARE`: This is used for a square hollow bullet.

Ordered List Tag Extensions: The following are the ordered list tag extensions:

- `TYPE=1`, this is the default which labels the list item with numbers (1, 2, 3).
- `TYPE=A`, this orders list items with capital letters (A, B, C).
- `TYPE=a`, this orders list items with lower case letters (a, b, c).
- `TYPE=I`, this orders the list items with capital roman numerals (I, II, III).
- `TYPE=i`, this orders the list items with lower case roman numerals (i, ii, iii).

NOTES

Check Your Progress

26. How an XHTML document is divided?
27. What is DHTML?
28. What does DHTML allow?
29. What is DOM?
30. What does HTML DOM define?
31. What is Internet Explorer? How it works?
32. What is Shockwave?
33. What are Netscape extensions? When these are avoided?

<FONT>...</FONT>

The `<FONT>` tag encloses text and prescribes the font size. This tag is controversial. By assigning font size, the Netscape extension begins to control the format of a document and may undermine the flexibility and portability of HTML. The values of font size that can be assigned are 1 to 7 as shown below:

```
Font 1 <FONT SIZE=1>Font 1</FONT
Font 2 <FONT SIZE=2>Font 2</FONT>
Font 3 <FONT SIZE=3>Font 3</FONT>
Font 4 <FONT SIZE=4>Font 4</FONT>
Font 5 <FONT SIZE=5>Font 5</FONT>
Font 6 <FONT SIZE=6>Font 6</FONT>
Font 7 <FONT SIZE=7>Font 7</FONT>
```

You can also specify font size in relative terms by using the '+' or '–' characters. The font will change relative to the default font size of your browser. This extension is used with utmost care otherwise it may cause problems for the users. For example, if a user has set their browser for 16 pitch Time Roman because they need larger print then the font with a minus value might cause them difficulty in reading the text and the positive values may cause the text to be too large on their screen.

Relative font size:

```
Font +3 <FONT SIZE=+3>Font +3</FONT>
Font -3 <FONT SIZE=-3>Font -3</FONT>
```

Backgrounds

The background attributes permits to add images as backdrops to your document, as well as manipulate the background color, text and link colors of a document. The most important thing to consider when using these tags is the readability of the documents. There are many examples of the improper use of these tags on the Web. Background attributes are inserted into the `<BODY>` tag as follows.

- **<BODY BACKGROUND="../image/purple.gif>:** This element tells the browser to find the file 'purple.gif' in the image directory and display it as a background.
- **<BODY BGCOLOR="#E5E2FF">:** This element tells the browser to display the background color as something other than the default background (grey). The BGCOLOR attribute can be used as an alternative to the BACKGROUND attribute if the users are running 14.4 modems. Other attributes that can be added are as follows:
  - TEXT: It controls the color of all the regular text in the document.
  - LINK: It controls the color of the normal unvisited links.
  - VLINK: It controls the color of the visited links.
  - ALINK: It controls the color of the link when the user presses the mouse button on the link

<BLINK>...</BLINK>

The `<BLINK>` tag cause the text to blink on and off. Depending on the version of Netscape the users are using, either the text itself will blink or a white block will blink on and off behind the text. This tag is used to call attention to a specific item.

NOTES

Check Your Progress

34. Fill in the blanks with appropriate words.
    (a) The client/application which submits an HTTP request to a server is referred as an ______________ .
    (b) The size value is an integer that specifies the font size in______________ point.
    (c) HTML colors are defined using a ______________ notation for the combination of RGB color values.
    (d) The multimedia objects can be installed by ______________ element that includes multimedia content with HTML document.
35. State whether the following statements are true or false.
    (a) One can also insert comments into the HTML file and these comments will not display on the browser.
    (b) Superscript text appears half a character below the baseline.
    (c) Transparency can be brought in an image using the style property of the filter.
    (d) An e-mail message consists of three components, the message envelope, the message header and the message body.

4.17 SUMMARY

NOTES

- In 1989, Tim proposed a hypertext system based on the Internet and by end of 1991, wrote the first specifications for HTML which contained 20 elements. Thirteen of these elements still exist in HTML 4. HTML specifications are maintained by the W3C (World Wide Web Consortium).
- HTTP uses TCP/IP for reliable transport, i.e., the transfer of Web pages from server to client over HTTP is done on TCP/IP layer. HTTP is not constrained to use TCP/IP, it can use any Internet protocol that provides reliable transport.
- For HTTP protocol, this matrix is unknown as it relies on TCP/IP for the connection between two machines in different networks. HTTP can work over any protocol that provides a reliable transport.
- HTML pages are written in the form of HTML elements. HTML elements are tags, surrounded by angle brackets '<' and '>', which act as indicators to a Web browser as to how, the document is to be interpreted by the browser and ultimately presented on the user's computer screen.
- Attribute `alink` is an attribute of the `<body>` tag. The value of this attribute is a color value specified either by its standard name or by its RGB value. It renders all the links present in the Web page of the specified color value.
- Tags which are used to format the edited text in the HTML document, such as `<del>` defines text that has been deleted from a document whereas `<ins>` defines text that has been inserted into a document.
- An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag. The list items are marked with bullets. An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag. The list items are marked with numbers.
- The `<a>` element is usually referred as a link or a hyperlink. The most important attribute of the `<a>` element is the `href` attribute which indicates the link's destination. By default, links will appear in all browsers.
- A URL's appearance, cookie preferences and default home page are controlled by your browser settings. A URL is a protocol that is used to identify a Web site. According to Web Developer Notes, a URL includes the protocol (HTTP), host name (WWW), domain name and domain type.
- The `<tr>` tag is used to define the row in an HTML table, and the tag `<td>` is used to define a column in the row inside a `<tr>` tag. It is recommended to keep the same number of columns in all rows otherwise some browser may not render the table properly.
- The border attribute is used to specify the border width of the table grid. The value as specified corresponds to the number of pixels to construct the grid. If this value is set to '0' then there will not be any visible border.
- XHTML means 'eXtensible HyperText Markup Language'. It can be interpreted as 'the stricter and cleaner version of HTML'.
- In case of XHTML, all elements must be nested within the `<html>` root element. All other elements in XHTML are sub-elements or children of the root element. Other elements can further have sub-elements.
- DHTML stands for Dynamic Hyper Text Markup Language or simply Dynamic HTML. It is a combination of HTML, CSS, JavaScript, and Document Object

NOTES

Model (DOM). DHTML can be described as the art of making interactive and dynamic HTML documents.

- The major disadvantage of DHTML is that it is difficult to debug and develop Web pages due to the difference in support as extended by Web browsers for scripting languages.
- DOM (Document Object Model) is an independent platform and a language neutral interface that allows the scripts and programs to access/update the document content, its structure, and style dynamically.
- According to DOM, everything in an HTML document is a node, even the text and comments are nodes, termed as 'Text Nodes' and 'Comment Nodes' respectively.
- Internet Explorer is a series of graphical Web browsers developed by Microsoft and included as part of the Microsoft Windows operating systems.
- Each of the browser available today consists of three parts, a controller, client protocol and interpreters, and so the Internet explorer.
- Shockwave is not a development tool but it is a runtime routine. There are three different versions of Shockwave. Two versions are runtime routines for multimedia applications and the third version is for a graphics package.
- Shockwave is a browser plug-in that enables the specific applications, such as Macromedia Inc., Macromedia Director, Macromedia Authorware and Macromedia Freehand to play on an Internet Web site or on an intranet.
- Lingo is a scripting language developed by John H. Thompson for Adobe Director (formerly Macromedia Director) specifically used for regular desktop applications, interactive kiosks, CD-ROMs and Adobe Shockwave Internet Websites.
- Netscape extensions focus primarily on making documents appear enhanced. They are an excellent method to improve the appearance and impact of a home page in an increasingly Netscape dominated Web.
- Netscape extensions should be avoided when publishing reports, articles and other items where the textual content is more important. They could cause problems when HTML 3 is finalized.

4.18 ANSWERS TO 'CHECK YOUR PROGRESS'

1. The Web Server which stores HTML files and resources is known as origin server.
2. Proxies are servers which act as a proxy for you over the network and hide the network behind it.
3. The HTML documents are delivered from a Web server to a client via the HTTP protocol running over TCP/IP.
4. All HTML file uses .html or .htm as file name extension. This extension uniquely identifies an HTML file and associates them directly to the Web browser.
5. HTML elements can be described as tags surrounded by angle brackets '<' and '>', which are interpreted by the browser to perform a specific formatting.
6. An HTML file has two sections HEAD and BODY. The HEAD part is described by content between the `<head>` and `</head>` tag, while the BODY part is described by the content between `<body>` and `</body>` tag.

NOTES

7. The <html> element is also known as the root element.
8. The <applet> deprecated tag is used as <object> replacement to insert applets in HTML.
9. One of the features of HTML tags is that HTML tags are not case sensitive. The World Wide Web Consortium or W3C recommends lowercase in HTML 4 and demands lowercase tags in the future versions of XHTML.
10. The attribute `bgcolor` is used to specify the background color of the Web document when it is displayed in the browser.
11. HTML tag <a> is used to insert a hyperlink in an HTML page.
12. The <face> attribute specified in the <font> tag renders the text in that font face.
13. The <size> attribute specified in the <font> tag renders the text in that particular font size.
14. In HTML, images are defined with the <img> tag. Basically, the <img> tag is empty, which means that it contains attributes only and has no closing tag. To display an image on a page, you need to use the `src` attribute, where `src` stands for 'source'. The height and width attributes are used to specify the height and width of an image.
15. Inline images are seen on a Web page with the text and links. They are loaded automatically when an HTML document is viewed on the Web browser. External images cannot be displayed directly. They are stored isolated from the HTML document and are loaded only when asked for Graphical Interchange (GIF) and Joint Photographic Experts Group (JPEG) are the generally used formats for images.
16. `align` specifies the image alignment in reference to the surrounding text.
17. A hyperlink or link is a reference to data that the reader can directly follow or that is followed automatically. It points to a whole document or to a specific element within a document. Hypertext is text which is used with hyperlinks. A software system is used to view and create hypertext called as hypertext system and also to create a hyperlink for the Web sites.
18. An anchor represents a string or text which marks the beginning and/or the end of a hypertext link. The text between the opening tag and the closing tag is either the start or destination (or both) of a link.
19. The <table> tag is used to define an HTML table.
20. The `align` attribute is used to align a table to right, left or center into the Web page.
21. The `cellpadding` attribute is used to specify the width of padding around a table cell in pixels. In other words it introduces a frame of width, specified in this attribute value around the cell.
22. The <caption> tag is used to add a caption row to the table.
23. HTML frames allow an HTML programmer to present the Web pages in different views.
24. The <frameset> tag is a replacement of the <body> tag which defines a partition in the browser window.

25. HTML forms are used to take different types of user inputs and pass on the data to the server.
26. An XHTML document can be divided into the following components:
    - DOCTYPE declaration.
    - <head> section.
    - <body> section.
27. DHTML stands for Dynamic Hyper Text Markup Language or simply Dynamic HTML. It is a combination of HTML, CSS, JavaScript, and Document Object Model (DOM). DHTML can be described as the art of making interactive and dynamic HTML documents.
28. DHTML allows the scripting languages to change the variables and attribute values of the element on screen, which in turns affects the look and feel of an HTML page. Otherwise, an HTML page is static after it is fully loaded into the browser window.
29. DOM (Document Object Model) is an independent platform and a language neutral interface that allows the scripts and programs to access/update the document content, its structure and style dynamically. Basically, the DOM defines the objects and properties for all the document elements and the methods to access them.
30. The HTML DOM defines a standard way for accessing and manipulating HTML documents.
31. Internet Explorer is a series of graphical Web browsers, developed by Microsoft and included as a part of the Microsoft Windows operating systems. Each of the browser available these days consists of three parts, a controller, client protocol and interpreters, and so the Internet Explorer. The controller receives input from input devices, such as keyboard, mouse, etc. It uses client programs to access the Web pages. After the document has been accessed, the controller uses one of the interpreters to display the document on screen.
32. Shockwave is a browser plug-in that enables the specific applications, such as Macromedia Inc., Macromedia Director, Macromedia Authorware and Macromedia Freehand to play on an Internet Web site or on an intranet. Although Shockwave is not a development tool, it does enable the applications that have been shocked to include Web-related commands, such as the loading of HTML pages and to become interactive tools for the World Wide Web.
33. Netscape extensions focus primarily on making documents appear enhanced. They are an excellent method to improve the appearance and impact of a home page in an increasingly Netscape dominated Web. Netscape extensions should be avoided when publishing reports, articles and other items where the textual content is more important.
34. (a) User agent, (b) Pixel, (c) Hexadecimal, (d) OBJECT.
35. (a) True, (b) False, (c) False, (d) True.

4.19 QUESTIONS AND EXERCISES

NOTES

Short-Answer Questions

1. What are the advantages of HTML?
2. What are the devices used over the Internet?
3. Name the HTML file name extension.
4. Write the main features of HTML.
5. What does `<font>` tag specify?
6. Write the function of `<sup>` tag.
7. What does `<dl>` define?
8. What is hyperlink?
9. What are anchors?
10. What is `cellspacing` attribute?
11. Why `colspan` attribute is used?
12. Name the tag which is used to create HTML frame.
13. Why submit button is used?
14. Write the two ways of associating style to an element.
15. Why text properties are used to set the behavior of the text?
16. What is XHTML?
17. What are the various components of an XHTML document?
18. What is the significance of DHTML?
19. Name some latest Microsoft Internet Explorers.
20. What is an access indicator?
21. What is Lingo?
22. When are Netscape extensions used?

Long-Answer Questions

1. Describe the basic tags of HTML with the help of examples.
2. How a simple page is designed? Explain with the suitable program code.
3. Discuss the process of creating an HTML document.
4. Differentiate between essential and deprecated tags with the help of examples.
5. Discuss the difference between `alink` and `vlink` with the help of example program codes.
6. How color and background are set for Web pages? Explain with the help of examples.
7. Discuss lists and their types using suitable examples.
8. Explain the attributes of `<image>` tag with the help of examples.
9. Write an HTML code to insert image on Website.
10. Describe the working of Web with the help of examples.
11. Discuss the common URL schemes used in HTML.

NOTES

12. Write a HTML code to create a table.
13. Explain the process of coloring table using `bgcolor` tag.
14. Write a HTML code to create a frame.
15. Discuss the significance of password field with the help of an example.
16. Why is XHTML used extensively nowadays? Explain with the help of an example.
17. Discuss the various characteristics of XHTML.
18. Explain the steps required to convert HTML to XHTML.
19. Discuss the advantages and disadvantages of using Dynamic HTML.
20. Explain the properties and methods as per DOM specifications.
21. Discuss the anatomy and features of a Web Browser toolbar.
22. Explain the significance of Microsoft Internet Explorer. List all the popular ones along with the version.
23. Discuss the significance of Shockwave and Lingo.
24. Explain the commands and functions used in Shockwave with the help of syntax and examples.
25. Discuss the three types of scripts used in Lingo.
26. Explain the important features of Netscape extensions.
27. Explain the significance of various Netscape extensions tags that are used in many Web documents along with the results.

UNIT 5 ACTIVEX CONTROLS

NOTES

Structure

5.0 INTRODUCTION

In this unit, you will learn about the basics of ActiveX Controls. Servers can be implemented as ActiveX DLL or ActiveX EXE components. The difference lies in how the server is executed. An ActiveX DLL is an in-process server. The DLL is loaded in the same address space as the client executable that calls the server and it runs on the same thread as the client. An ActiveX component is basically a user defined executable control that can be used to create interfaces and applications like other controls, such as `TextBox` and `CommandButton`. ActiveX components allow programmers to reuse these user defined controls to perform manipulation on various data types in their applications. ActiveX components are saved as a file with `.exe`, `.dll` or `.ocx` extensions. Finally, you will learn about Web based mail which is a implemented as a Web application accessed via a Web browser.

5.1 UNIT OBJECTIVES

After going through this unit, you should be able to:

- Understand the basics of ActiveX EXE and ActiveX DLL
- Explain ActiveX controls
- Define the various steps required for creating ActiveX control
- Describe the significance of creating an ActiveX control to activate a Web page

- Explain VDOLive technology
- Define how to create Netscape Navigator plug-ins
- Learn about pulling Web information

NOTES

5.2 ACTIVEX: BASICS AND CONTROLS

An ActiveX control is unique in its ability to be contained within something else. A familiar example is a control contained within a Visual Basic form. ActiveX controls are not limited to Visual Basic. However, more and more Windows development tools support ActiveX; some well-known examples are Delphi, Visual C++ and PowerBuilder. Although these tools differ from Visual Basic in many respects, they share the ability to drop ActiveX controls onto a form, providing all of the control's functionality, with little or no programming.

5.2.1 ActiveX EXE and ActiveX DLL

Servers can be implemented as ActiveX DLL or ActiveX EXE components. The difference lies in how the server is executed. An ActiveX DLL is an in-process server. The DLL is loaded in the same address space as the client executable that calls the server and it runs on the same thread as the client. At any given moment however, the client application or the DLL keeps running. The merit of DLL is that it is faster as, in effect, it becomes a part of the application that uses it.

An ActiveX EXE, otherwise called out-of-process server, runs as a separate process. When a client application creates an object provided by an EXE server for the first time, the server starts running as a separate process. If another client application creates the same object, the running EXE server provides this object. In other words, a single EXE server can service multiple clients. Out-of-process servers seem to be more efficient in terms of resource allocation, but exchanging information between servers is a slow process.

We will be working with a client–server model. Here the application is the client, which sends a request, and the ActiveX component is the server, which services the request. Though ActiveX DLLs run within the same application as the client, they are considered to be servers.

The distinction between server types results in differences in testing the components. An ActiveX DLL can be tested by adding a test project to the ActiveX DLL project. Since both the server components and the test application run in the same process, we don't need two different executables.

Creating and Compiling an ActiveX Component

An ActiveX component can be created in Visual Basic by starting a new project. The ActiveX EXE or ActiveX DLL is chosen depending on the type of the project to be created. A new project is created with a single class module. If needed, additional classes can be included to the project. Suitable coding is written for the classes. The final step is to compile the project into an ActiveX DLL or EXE.

An ActiveX project is compiled in the same way as the standard EXE project. But ActiveX EXEs and DLLs are used differently. Thus, ActiveX components are Object Servers that can be used with other applications.

While both ActiveX DLLs and EXEs can provide objects to other applications, ActiveX EXEs have the capability to execute independently, which is not so in ActiveX DLLs.

Compilation of an ActiveX component can be done by selecting File->Make menu command as with a standard EXE project. The project is then compiled into an ActiveX DLL or EXE as appropriate. After it is compiled, the component is registered on the computer, so that the user can use the objects created from its classes in other applications.

Let us now move on to creating ActiveX EXE and ActiveX DLL.

NOTES

Creating an ActiveX EXE Component

An **ActiveX EXE** component is an out-of process server which can be developed and run independently. It can be included in the client application after it is compiled and registered. The procedure of creating an ActiveX EXE application, compiling and registering the same and then testing it using a client application is elucidated here.



ActiveX EXE: An out-of process server which can be developed and run independently

Note: The ActiveX server can be used not only by any Visual Basic project, but also by any development environment that knows about ActiveX components.

Creating an ActiveX DLL Component

ActiveX DLLs (Code Components) (In Process)

An ActiveX DLL is an in-process server, which runs as a part of the client application. When an ActiveX DLL component is created, the following steps are generally followed:

- Determining the features that the component will provide.
- Determining what objects are required to divide the functionality of the component in a logical fashion.
- Designing the forms that the component will display.
- Designing the interface, that is, the properties, methods and events for each class provided by the component.
- Creating a project group consisting of the user's component project and a test project.
- Implementing the forms required by the component.
- Implementing the interface of each class.
- Adding each interface element or feature to the test project to exercise the new functionality.
- Compiling the `DLL` and testing it with all potential target applications.

Components provide reusable codes in the form of objects. An application that uses a component's code by creating objects and calling their properties and methods is referred to as a client. Components can run either in-process or out-of-process with respect to the clients who use the objects. An in-process component or ActiveX DLL, runs in another application's process. The client may be the application itself, or another in-process component that the application is using.

The advantages of DLLs are:

- The code can be easily shared among applications.
- They offer excellent performance due to the in-process nature of the component.

NOTES

- Fixing a bug in a DLL Implement object only requires distributing an updated DLL.
- All applications using the DLL are immediately fixed.
- They can be used by any OLE automation client, including all VBA-based applications (such as Microsoft Office) and other Windows development languages.

The disadvantages of DLLs are:

- If an updated DLL is incompatible with its predecessor, you can break every application that uses the DLL.
- It does not support multithreaded objects in VB 5.0.
- It increases the complexity of deploying an application.
- It requires registration, version checking and component verification for safe distribution.

To summarize, it is ideal for implementing standard objects that you may wish to reuse or share among applications. It is also ideal for defining the interfaces to be implemented by other objects. And it is the preferred way to create high-performance objects that do not have a user interface.

A series of step-by-step procedures in this unit builds an in-process component called Interest Calculator with class modules that demonstrate object lifetime.

5.2.2 ActiveX Controls

Of course, as a programmer, you need to be able to create ActiveX controls, not just use them. Using ActiveX controls is not a major challenge. From the Visual Basic programmer's perspective, they closely resemble non-ActiveX controls. In both cases, the control appears in the Visual Basic toolbox; it has properties, methods, events, and so on. The CheckBox control, for example, is an 'old-fashioned' OCX control, while the TreeView control is an ActiveX control. ActiveX controls do have capabilities not found in the older OCX controls, but these features are not immediately obvious.

Creating an ActiveX control is a relatively simple task in Visual Basic. For the most part, the procedure is the same as creating a standard Visual Basic executable. You start with a form, place controls on it (including other ActiveX controls) and then write code to define properties and methods and deal with events. When you compile the project to an OCX file, Visual Basic takes care of all the details. And yes, ActiveX controls are saved in files with a .ocx extension, just as the older OCX controls. The Windows registry makes information about the capabilities of a given OCX file available to potential container programs.

An ActiveX control does not have to contain other controls. You can draw the control using the various graphics methods in Visual Basic, such as **Line** and **Circle**.

Advantages of ActiveX Control

There are many advantages in using ActiveX controls:

- Good performance. ActiveX controls always run in process. However, there is an additional overhead involved in using an ActiveX control that does not occur with an ActiveX DLL server.

NOTES

- Controls are compatible with many containers, including Microsoft Office applications and the Internet browsers.
- Controls offer seamless integration into the VB environment.
- Property pages allow design time user interface as well as run-time interface within Visual Basic.
- Controls have the ability to persist design time properties in most containers.

Disadvantages of ActiveX Control

The disadvantages of ActiveX control are:

- Controls are considerably faster than ActiveX EXE servers, but somewhat slower than ActiveX DLL servers due to the ActiveX overhead.
- There is some complexity involved in creating good quality controls.
- Controls increase the complexity of deploying an application.
- Registration, version checking and component verification are required for safe distribution.

To summarize, controls are ideal for implementing reusable objects that have a user interface. They are useful in many cases for improving the modularity of an application.

Creating ActiveX Control

In this section, you will be taken through the process of creating, testing and using an ActiveX control. The control will be relatively simple as my goal is to present the procedures that are specific for creating ActiveX controls. What goes in the control does not differ from 'regular' Visual Basic programming. In other words, the functionality of an ActiveX control is created in essentially the same way as the functionality of a standard Visual Basic executable—you place controls on a form, write event procedures, manipulate properties, and so on. You will also find some elements in common with creating other ActiveX objects, specifically the use of *property* procedures to define properties for your object (in this case, the control you are creating).

What will the demonstration control do? Its name, `FancyCmdButton`, describes it well. It will serve the same function as a regular Visual Basic Command Button, which the user can click on to trigger an action. It will have a slightly more appealing appearance: a coloured background that changes to indicate that it has been clicked. Most ActiveX controls are significantly more complicated than this. However, our example is ideal for demonstrating the major parts of creating and testing an ActiveX control.

Many of the procedures involved in creating our ActiveX control could be performed using the Class Builder utility. You will be taken through each step without using the utility, because it will be a better way for you to learn the nuts and bolts of creating an ActiveX control. Once you understand the procedure, then you can use the Class Builder utility for your future projects to save time and effort.

Start by firing up Visual Basic and selecting New Project from the File menu. Select ActiveX Control from the available project types. Visual Basic starts the new project and adds a `UserControl` designer to it. (A `UserControl` object is the foundation of all ActiveX controls you create in Visual Basic). Your screen will look like as shown in Figure 5.1.

NOTES

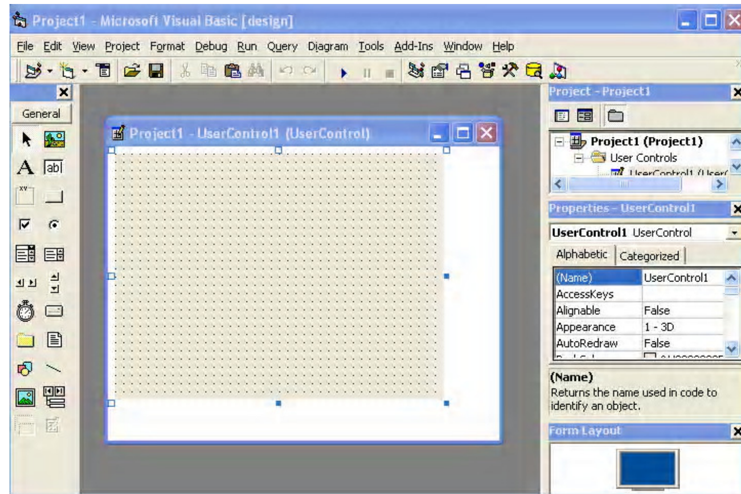


Fig. 5.1 The Visual Basic Screen after starting a New ActiveX Control Project

You should note the following elements:

- The `UserControl` Designer window displays the default project name (Project1) and control name (`UserControl`) in its title bar. This will change once we assign meaningful names to the project and control.
- The gray rectangle in the window represents the new control. It is similar to a standard Visual Basic form in some respects, but it does not have a border or title bar (although you can add a border).
- The Form Layout window does not show the control. The control's display position is determined by its container, not by any properties you set during design.

The next step is to set some of the project properties. Select Project1 Properties from the Project menu to display the Project Properties dialog box (see Figure 5.2). Display the General tab, make the following entries and then close the dialog box:

- Project name—`AXCTRLDEMO` is the name that will be assigned to the final compiled OCX file.
- Project description—Fancy Command Button is the description users will see when they use the OCX control in other projects.

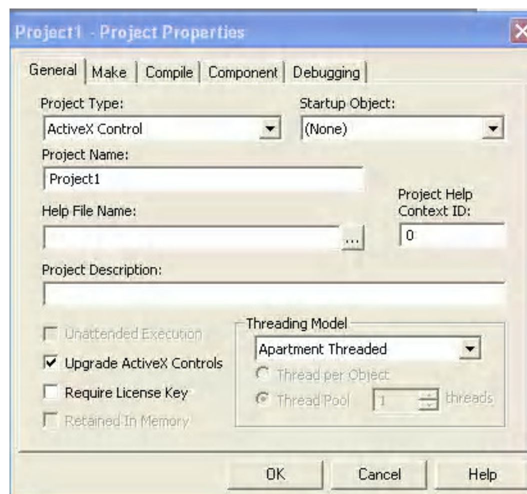


Fig. 5.2 The Project Properties Dialog Box

NOTES

Next, you need to change the Name property of the control itself. Be sure that the control is selected in the Designer window as indicated by handles around its periphery. In the Properties window, find the Name property and change it to `FancyCmdButton`. This name will now appear in the designer's title bar.

Now is a good time to save the project. Select Save Project from the File menu, saving both the control and the project under the default names that Visual Basic suggests: **Fancycmdbutton.ctl** and **Axctrldemo.vbp**.

Although the basic framework of the control is in place, it still doesn't do anything. To add functionality, start by placing a Shape control in the upper left corner of the control. Set its properties as shown in Table 5.2.

Next, add a Label control on top of the Shape control, setting its properties as shown in Table 5.3. Then, reduce the size of the `UserControl`. The exact size is not crucial—just keep it fairly small. At this stage, your screen will look as shown in Figure 5.3.

Table 5.2 *Shape Control Property Settings*

Property Name	**Setting**
BorderStyle	0 — Transparent
FillColor	&H0000FF00& (or any light green)
FillStyle	0 — Solid
Shape	4 — Rounded Rectangle
Nameshp	Button

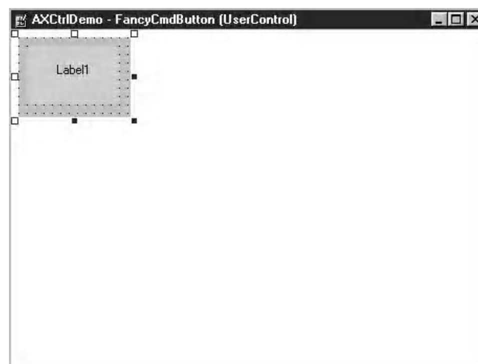


Fig. 5.3 *The `FancyCmdButton` ActiveX Control after Adding Shape and Label Controls*

Table 5.3 *Label Control Property Settings*

Property Name	**Setting**
BackStyle	0 — Transparent
Name	lblButton
ForeColor	Black
Alignment	2 — Center

The `FancyCmdButton` control now has all of the subcomponents (other controls) it needs, but that's not enough, of course. What will it do? How will it appear? Now we need to add the code that will bring our ActiveX control to life.

Our first concern is its appearance. When displayed within a container, we want the Shape control to fill the entire area of the ActiveX control; we also want the Label to be the same width as the Shape control and to be centred vertically. The code to perform these actions should be placed within the `UserControl's` **Resize**

event procedure. Display the Code Editing window in the usual manner and then select `UserControl` in the object list at the top of the window. Finally, select Resize in the Procedure list. Add the code in Listing 5.1 to this event procedure.

NOTES

Listing 5.1 The `UserControl's Resize` Event Procedure

```
Private Sub UserControl_Resize()
```

- Change the position and size of the Shape control to fill the `FancyCmdButton` control's entire area. shpButton.Move 0, 0, ScaleWidth, ScaleHeight
- Make the Label control the same width as the Shape control and center it vertically.

```
lblButton.Move 0, (ScaleHeight - lblButton.Height) /
2, ScaleWidth

End Sub
```

Testing ActiveX Control

The control is not finished, but it is ready to be tested. How do you test an ActiveX control that you are developing? An ActiveX control cannot run by itself; it needs a container. You have two choices:

- *You can use Microsoft Internet Explorer or another ActiveX-capable browser as the test container.* After all, one important use for ActiveX controls is on Web pages; so in some situations, testing your control in a browser is perfectly appropriate. If you have not created a separate test project (as described later) and the ActiveX control project is the only one loaded into the Visual Basic environment, then 'running' the project will start Internet Explorer and display an instance of the ActiveX control. The control must be compiled first, as described later in the section titled 'Compiling the ActiveX Control'. You can try out the control's capabilities, as well as modify the associated HTML file if you want to test the control in combination with some scripting language elements. These procedures are beyond the scope of this book, although I will touch on them later in the unit when I cover using ActiveX controls on the Web.
- *You can create a separate Visual Basic project to test the control*. This is the preferred testing method because it provides greater flexibility. Creating a test project is our next topic.

Creating a Test Project

To test an ActiveX control, you need a separate Visual Basic project with a form on which you can place an instance of the control. Rather than starting a separate copy of Visual Basic, you can use one of Visual Basic's handier features: You can define a *project group* that contains two or more independent projects. In this case, the ActiveX control will be one project, and the second project will be a standard Visual Basic executable to test the ActiveX control.

To create a project group, select Add Project... from the File menu, then select Standard EXE as the project type. You will now have two designers open: one for the ActiveX control project and one for the Standard EXE project you just created. Both projects will be listed in the Project window, as shown in Figure 5.4. Next, save the project group by selecting Save Project Group from the File menu. Use the file names given here (the extensions are added automatically by Visual Basic):

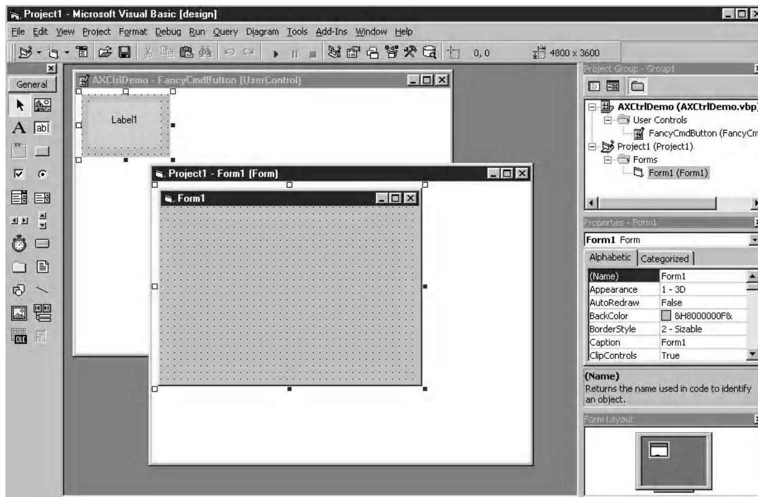


Fig. 5.4 A Visual Basic Project Group Can Contain Two or More Independent Projects

- *Form*—`TestAXCtl_Form1.frm`
- *Project*—`TextAXCtl.vbp`
- *Project group*—`TestAXCtl.vbg`

Running ActiveX Control at Design Time

An ActiveX control does not run in the same sense as a standard Visual Basic executable. When an ActiveX control 'runs', it makes itself available for insertion into other projects. This is exactly what we want: to run the ActiveX control, so that it becomes available to be inserted onto the test project's form, while the test project remains in the design mode. This is precisely what Visual Basic project groups are intended for.

The way it works is simple. Begin by closing the ActiveX control's designer by clicking on the close button (the X) in its title bar. This puts the ActiveX control into the run mode. The only sign of your ActiveX control now is the `UserControl` icon on the Toolbox (see Figure 5.5). Next, make the test project's form active and double-click on the `UserControl` icon to place an instance of it on the form as shown in Figure 5.5.

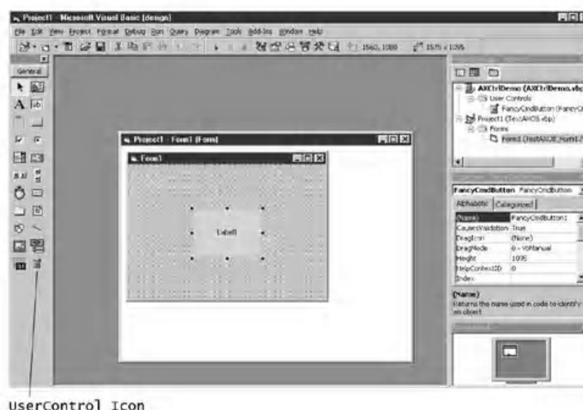


Fig. 5.5 The Test Project Form after Placing an Instance of the `UserControl` on It

Look in the Properties window: The new control instance has been given a default name, `FancyCmdButton1`, and it has its own set of properties. Change

NOTES

the **Name** property to FCB1 (for the sake of brevity). To take the ActiveX control out of the run mode and back into the design mode, double-click on its name in the Project window. When the ActiveX control designer opens up again, you will see that the instance of the control on the test project form becomes hatched, indicating that the control is no longer active. This is shown in Figure 5.6.

ActiveX Control Events

You will discover one difference between ActiveX controls and standard executable programs: An ActiveX control has a defined **behaviour not only during execution (when it is executing within its container) but also at design time.** To illustrate this, open the `UserControl's Resize` procedure and add the following line to the existing code:

```
Debug.Print "Resize event"
```

Close the ActiveX control designer to run the ActiveX control. On the test project form, change the size of the `FancyCmdButton` control. You will see by the messages displayed in the Immediate window that its `Resize` event procedure fires each time you resize the control, even though the test project is still in the design mode. If you add a second instance of the `FancyCmdButton` control to the form, you will see that the `Resize` event fires then, too. Delete the second control—if you placed one—before continuing.

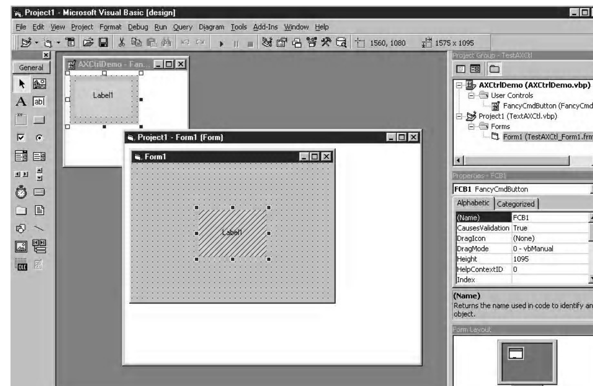


***Fig. 5.6** When the ActiveX Control is Placed Back in the Design Mode, the Instance on the Test Project Form displays a Hatch Pattern*

Now consider some of the other events that occur during the life of an ActiveX control. When working with these events, you must understand the ephemeral nature of an ActiveX control. You may think that once an ActiveX control is placed on a form, that's it; the control is created and continues to exist as a component of the form from then on. Things are not so simple.

When you place an ActiveX control on a Visual Basic form during program design, an instance of the control is created in memory. When you run the program, that instance is destroyed and a new run-time instance is created; it is this instance that will be in operation as the program executes. When you terminate the program to return to the design mode, the run-time instance of the control is destroyed, and yet another instance is created and displayed on the form in the Visual Basic designer. As you can see, rather than having a single instance of the control remaining, three have actually been created and destroyed.

For example, suppose you have designed an ActiveX control in Visual Basic and created a test form. When you close the ActiveX designer and place an instance of the control on the test form, you create an instance (as described in the previous paragraph). If you then reopen the ActiveX designer so that the control on the test form is displayed with hatching, the control instance is destroyed. When you close the ActiveX designer, a new instance of the control is created and the control on the test form is displayed without hatching.

NOTES

If you place two or more instances of a control on a form, each instance undergoes its own creation–destruction–creation cycle.

It is rather complicated. You will feel more comfortable with it after a while, particularly after some programming techniques are presented later in this unit that will assist you in keeping track of the creation and destruction of control instances. For now, what you should remember is that an ActiveX control has certain events that are triggered in response to control creation and destruction. Some of these events have to do with the control's properties—a topic we will be discussing later. For now, let us just look at the events without worrying about the details:



Initialize: Occurs each and every time an instance of the control is created. It is always the first event that occurs for the control

- ***Initialize***—Occurs each and every time an instance of the control is created. It is always the first event that occurs for the control.
- ***InitProperties***—Occurs only when the control is placed on a form; in other words, the first time an instance is created. You can use this event procedure to set a control's initial values.
- ***ReadProperties***—Occurs the second and subsequent times an instance of a control is created. In other words, this event occurs every time a control instance is created *except* the first time (when `InitProperties` occurs instead).
- ***Resize***—Occurs every time a control instance is created and every time its size is changed, either by the programmer in the design mode or by the code during program execution.
- ***Paint***—Occurs whenever the control's container tells the control to redraw itself. If you are creating your control's visual appearance with drawing methods, the code should go in this event procedure.
- ***WriteProperties***—Occurs during design time when an instance of the control is destroyed and at least one of its properties has been changed.
- ***Terminate***—The last event to occur when a control instance is destroyed.

To get a handle on when these various control events occur, you can use the same technique that you used earlier for the `Resize` event procedure: Place a `Debug.Print` statement in each event procedure to print the appropriate message to the Immediate window. The code is shown in Listing 5.2. With this code in your ActiveX project, you can trace events as they occur during the lifetime of your ActiveX control.

Listing 5.2 Using the Debug.Print Statement to Track the Occurrence of Event

```
Private Sub UserControl_Initialize()
    Debug.Print "Initialize event"
End Sub
Private Sub UserControl_InitProperties()
    Debug.Print "InitProperties event"
```

```
End Sub
Private Sub UserControl_Paint()
    Debug.Print "Paint event"
End Sub
Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
    Debug.Print "ReadProperties event"
End Sub
Private Sub UserControl_Terminate()
    Debug.Print "Terminate event"
End Sub
Private Sub UserControl_WriteProperties(PropBag As
PropertyBag)
    Debug.Print "WriteProperties event"
End Sub
```

NOTES

Responding to Events

When dealing with ActiveX controls, you have to consider events occurring at three levels:

- The constituent controls used to create the ActiveX control, such as the Shape and Label controls in the demonstration project, may need to respond to events.
- The ActiveX control may need to respond internally to events, such as the user clicking on the control.
- The container object may need to respond to events that occur to the ActiveX control.

If you think about this for a moment, you can see that events may need to be 'passed along' from one object to another. For example, if the user clicks on the Label control in the demonstration ActiveX control—and you want the container object to be able to respond—pass the event 'up' to the container. This is accomplished using Visual Basic `RaiseEvent` statement. The syntax is as follows:

```
RaiseEvent EventName [(ArgList)]
```

`EventName` is the name of the event to fire and `ArgList` is an optional list of arguments. Before you can use `RaiseEvent`, you must declare an event procedure for the event you will raise. This declaration must be at the module level of the module in which the event will be raised; it takes the following form:

```
[Public] Event EventName [(ArgList)]
```

`EventName` and `ArgList` are, respectively, the name of the event and an optional argument list. Include the optional **Public** keyword if the event needs to be detected by another module; otherwise, it will be available only within the module where it is raised. When the event is raised, the argument list used in the `RaiseEvent` statement must match the list in the event procedure declaration.

What events are available? The usual repertoire of Visual Basic events is at your disposal, such as `Click` and `MouseDown`. Without `RaiseEvent`, the control would be able to use events internally, but if you want the container to be able to respond to events, you will have to raise them. Before we write the event code for the `FancyCmdButton` control, let's think for a moment about what it needs to do:

- When the user presses the mouse button when the pointer is on the control, the control's background colour should change.

- When the user releases the mouse button, the background colour should change back, and a `Click` event should be raised so that the container object can respond to it.

NOTES

For the first task, we will use the `MouseDown` event. But where will this event be detected? The ActiveX control consists of both Label and Shape control, plus the underlying `UserControl`. Clearly, the Label control must respond to `MouseDown`. Shape controls do not detect mouse events, so mouse action on our Shape control will be automatically passed through to the underlying `UserControl`. Thus, the `UserControl`'s `MouseDown` event procedure will also be used. In figuring out how to handle the first task, we learn how to do the second as well. We will use the `MouseUp` event procedure of the Label control and `UserControl`.

Start by opening the Code window for the `FancyCmdButton` control. Select General in the Object list and Declarations in the Procedure box and then add the code shown in Listing 5.3. This code declares a variable and a constant for manipulating the control's colour and declares the `Click` event procedure, so that we can use the `RaiseEvent` statement.

Listing 5.3 Code in the General Declarations Section of the ActiveX Control Module

```
Option Explicit
'Variable for the old color.
Dim OldColor As Long
'Constant for the "clicked" color (this is red).
Const NEWCOLOR = &HFF&
'Declare a Public Click event procedure.
Public Event Click()
```

The next code must be added to the `MouseDown` and `MouseUp` event procedures of the Label and `FancyCmdButton`. In the Code Editing window, use the Object and Procedure lists to select these procedures and then add the code shown in Listing 5.4. This listing combines the code for the two `MouseDown` and two `MouseUp` event procedures.

Listing 5.4 The MouseDown and MouseUp Event Procedures for the `FancyCmdButton` and the Label Control

```
Private Sub lblButton_MouseDown(Button As Integer, Shift As
Integer, X As Single, Y As Single)
'Save the original fill color.
OldColor = shpButton.FillColor
'Change to the "clicked" fill color.
shpButton.FillColor = NEWCOLOR
End Sub
Private Sub lblButton_MouseUp(Button As Integer, Shift As
Integer, X As Single, Y As Single)
'Restore the original fill color.
shpButton.FillColor = OldColor
'Raise the click event.
RaiseEvent Click
End Sub
```

NOTES

```
Private Sub UserControl_MouseDown(Button As Integer, Shift
As Integer, X As Single, Y As Single)
'Save the old fill color.
OldColor = shpButton.FillColor
'Change to the "clicked" fill color.
shpButton.FillColor = NEWCOLOR
End Sub
Private Sub UserControl_MouseUp(Button As Integer, Shift As
Integer, X As Single, Y As Single)
'Restore the original fill color.
shpButton.FillColor = OldColor
'Raise a Click event.
RaiseEvent Click
End Sub
```

This completes the code required to have our `FancyCmdButton` control respond to user clicks by changing its background colour and raising a click for its container to respond to. The next task is to add the code to the test project, enabling it to respond to that event. Close the `UserControl`, and display the Code window for the test project form (`TestAXCtl_Form1`). Add the single line of code in Listing 5.5 to the form's `Click` event procedure.

Listing 5.5 The Container Form's Click Event Procedure

```
Private Sub FCB1_Click()
MsgBox ("I've been clicked")
End Sub
```

The project is now ready to be taken for a spin. Be sure that the FancyCmdButton designer is closed, as indicated by the control on the test form displayed without hatch marks. Also, be sure that the test project is the startup project, as indicated by its name displayed in bold in the Project Explorer window. If it is not, right-click on the project name and select Set As Startup from the pop-up menu. Then, press F5 to run the test project. You will see its form displayed, as shown in Figure 5.7. When you position the mouse pointer over the control and press the mouse button, you will see the button's background colour change to red. When you release the mouse button, the colour changes back to green and a message appears indicating that the form has detected the click.

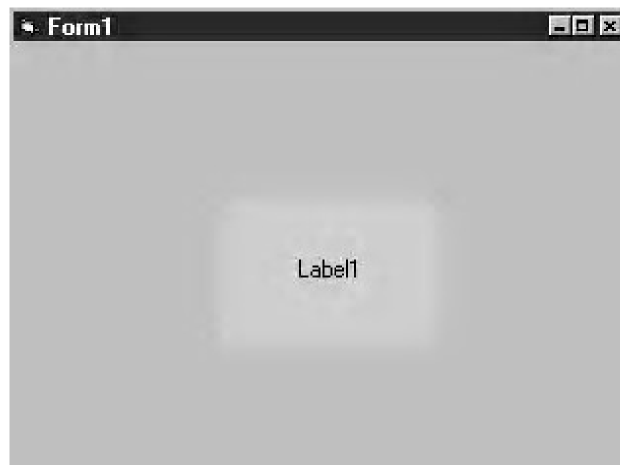


Fig. 5.7 *Testing the* `FancyCmdButton` *Control*

> *Note:* **ActiveX Testing Options**
>
> Even if you have created a test project for your ActiveX control, you have the option of testing the control in the Internet Explorer Web browser. If the test project is set as the startup project, the control will run in your test project. If you make the ActiveX control project the startup, then the control will run in Internet Explorer.

NOTES

Before we continue adding to the demonstration control, think for a moment about what happens when you execute the test program and click on the `FancyCmdButton` control. The `MouseDown` and `MouseUp` events are detected by the ActiveX control itself, which responds by changing its background colour. In addition, a `Click` event is raised, and that event is detected by the container—the test program. In response to that event, a message box is displayed.

Adding Properties to Control

Properties are added to an ActiveX control in the same manner as for any other ActiveX object. To define a property, you create property procedures. The `Get` procedure makes the property available for reading and the `Let` procedure is used to set the property value. To define a read only property, you can create a `Get` procedure without a corresponding `Let`.

We will create a single property for the `FancyCmdButton`. It will be called **Caption** and it will determine the text that is displayed on the control. Because the value will be stored in the Label control's `Caption` property, we do not need to declare a separate variable to hold it.

Here are the steps to be followed:

- Open the `FancyCmdButton` designer and then open its Code window.
- Select Add Procedure from the Tools menu to display the Insert Procedure dialog box.
- Specify the procedure name `Caption` and select the Property and Public options. Then click on OK.
- Visual Basic will create skeletons of the `Let` and `Get` procedures for you, displaying them in the Code Editing window. Add the code shown in Listing 5.6. Be sure to change the type of the `Get` procedure from `Variant` to `String`. Make the same change for the type of the argument to the **Let** procedure. Remember that the `Get` procedure's return type must be the same as the type of the `Let` procedure's argument.

Listing 5.6 Get and Let Procedures for the `Caption` Property

```
Public Property Get Caption() As String
    Caption = lblButton.Caption
End Property

Public Property Let Caption(ByVal vNewValue As String)
    lblButton.Caption = vNewValue
End Property
```

Once you have added this code, close the ActiveX designer to put the `FancyButtonControl` in the run mode. Display the test form with the `FancyCmdButton` on it and click on the button to select it. Look in the Properties window (see Figure 5.8), and you will see that the control's property list now includes

NOTES

a Caption property—the one you just defined—in addition to the UserControl's default properties. If you change this property during design, the text you specify will be displayed on the FancyCmdButton when the test project runs. You could also set the property in code. To try this out, place the following line of code in the test project's Form_Load procedure:

```
FCB1.Caption = "Click Me"
```

You will see that the button displays 'Click Me' when the program runs.

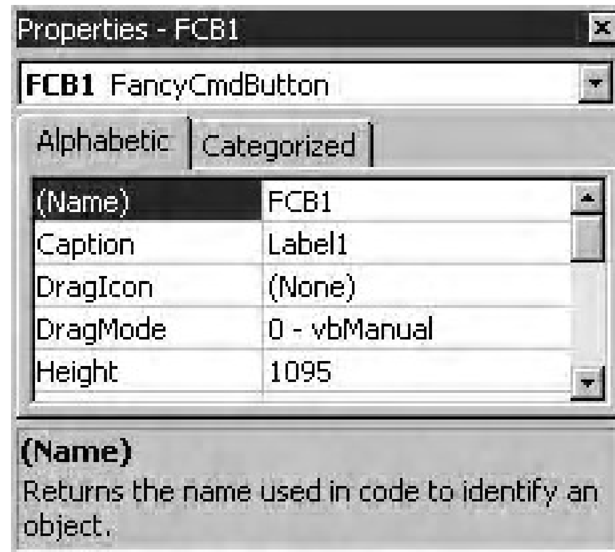


Fig. 5.8 *ActiveX Control Properties in the Visual Basic Properties Window along with the Default Properties*

Adding a Property Page to Control

You have already seen that the ActiveX control properties you define are automatically displayed in the Visual Basic Properties window. You also have the option of connecting a *property page* to the control. A property page is simply a different method of displaying and accessing the control's properties. Each property page you define will become a separate tab in the object's Properties dialog box. You must design the page, which is done in much the same way as designing a Visual Basic form. Visual Basic takes care of all the details of displaying the tabs, and managing the OK, Cancel and Apply buttons.

Property pages are useful when several properties interact in a complex fashion. You can design a property page so that related properties are grouped together, making it easier for the user to set them properly. Property pages are useful for controls that you plan to distribute internationally, because captions on property pages can easily be changed to suit different language requirements. Finally, property pages permit controls to be used with development tools that don't have Properties windows.

To add a property page to the FancyCmdButton, click on AXCtrlDemo in the Project window to make the control project current. Then, select Add Property Page from the Project menu. In the next dialog box, select the Property Page icon. You can also explore the other option—the Visual Basic Property Page wizard—on your own to creat unique options. Visual Basic adds a property page to the project as shown in Figure 5.9. The Property Page form is displayed and its properties are listed

in the Properties window. Note that the Visual Basic title bar indicates that the Property Page designer is active by displaying [PropertyPage1 (PropertyPage)]. Note also that a Property Page entry has been added to the listing in the Project window.

NOTES

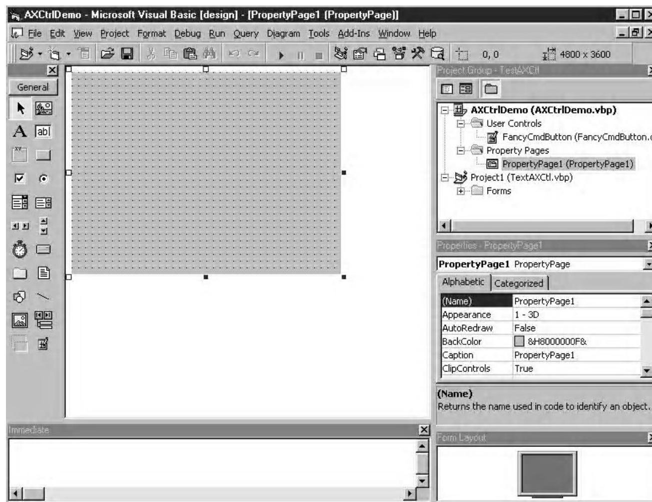


Fig. 5.9 *After Adding a Property Page, a Blank Property Page is Displayed*

In the Properties window, change the property page's `Name` property to FCBGeneral and the `Caption` property to General. The caption will be displayed as the tab title in the Properties dialog box; the name identifies it as the `FancyCmdButton`'s General property tab. Select Save from the File menu and save the property page under the suggested name, which is the same as the `Name` property you just assigned (FCBGeneral). Visual Basic automatically adds the .pag extension to property page files.

The next task is to design the property page itself, placing controls on it to permit the user to read and set the control's properties. Because the `FancyCmdButton` control has only a single property, `Caption`, this will be a quick task. Designing a property page is essentially the same as designing a regular Visual Basic form: drag and drop controls and so on.

Start by placing a Label control on the property page. Set the Label's `Caption` property to `Caption`. Place a Text Box control under the Label, and set its `Text` property to a blank string and its `Name` property to `txtCaption`. At this point, your property page will look like as shown in Figure 5.10.

The property page interacts with the control it is attached to by using events. Whenever a property page is opened, it receives a `SelectionChanged` event. It receives the same event if and when the user changes the controls that are selected on the current form (remember, the property page will be used when the user is designing a form and has placed one of your controls on it). Our task is complicated by the fact that the user can select more than one control—it is absolutely possible for a user to place two or more `FancyCmdButton` controls on a form and select all of them. Because a property page is modeless, the user can change the selected controls while the property page remains open.

NOTES

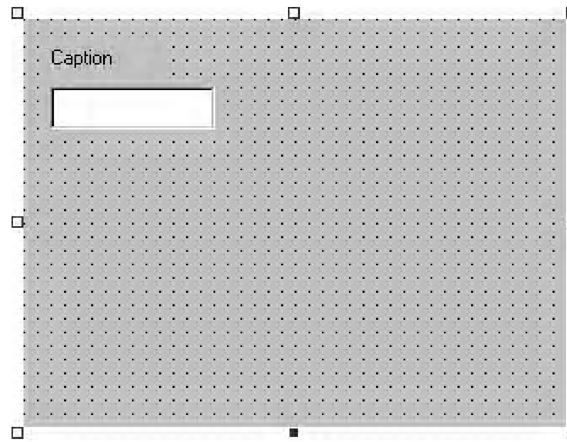


Fig. 5.10 *The Property Page after Placing its Two Controls*

Basically, you have two options for dealing with multiple selected controls. The one you use will depend on the nature of the specific property. For some properties, such as `ForeColor`, it makes sense to permit the user to change the property setting for two or more controls at once. Note that `ForeColor` is being used here as a generic example—it is not a property of our `FancyCmdButton` control. In contrast, other properties are not appropriate for such batch changes; if multiple controls are selected, you want to disable that property. The `Caption` property of our demonstration control falls into the latter category.

Dealing with the possibility of multiple selected controls is simplified by the `SelectedControls` collection, which provides a zero based index list of the control(s) that are currently selected on the form. You can query this collection's `Count` property to see if more than one control is selected and then take the appropriate action. For the single property on the `FancyCmdButton`'s property page, use the code shown in Listing 5.7. This code is placed in the property page's `SelectionChanged` event procedure.

Listing 5.7 The Property Page's `SelectionChanged` Event Procedure

```
Private Sub PropertyPage_SelectionChanged()
'Enable the Text Box for the Caption
'property only if there is a single
'control selected.
If SelectedControls.Count = 1 Then
    txtCaption.Enabled = True
    'Display the current property value on the property
page.
    txtCaption.Text = SelectedControls(0).Caption
Else
    txtCaption.Enabled = False
End If
End Sub
```

The code we have written takes care of displaying the current property value on the property page when the page is opened. It also disables the Text Box on the property page if more than one `FancyCmdButton` control is selected. We still have to write the code that moves information in the other direction: from the property page to the control's actual properties. This code is divided into two parts.

NOTES

First, every property page has a `Changed` property. You need to write a code that sets this property to True if the user makes any changes to the properties listed on the page. When its `Changed` property is True, the property page automatically enables its Apply button, which the user clicks on to apply the new properties to the control. The ideal place to do this is in the Text Box's `Change` event procedure, which is fired whenever the user changes the contents of the Text Box. To add this code, be sure that the property page is displayed (if not, double-click on FCBGeneral in the Project box). Then double-click on the single Text Box on the page to display the code for its `Change` event procedure and add the code shown in Listing 5.8.

Listing 5.8 The Change Event Procedure for the Text Box on the Property Page

```
Private Sub txtCaption_Change()
    'Set the property page's Changed
    'property to True if the user
    'changes the contents of the Text Box.
    Changed = True
End Sub
```

Second, to apply the change, use the property page's `ApplyChanges` event. This event is fired when the user clicks on either the OK or the Apply button in the Property Page dialog box. Your job is to place code in this event procedure that will copy property values from the controls on the property page to the actual control properties. The details of how this is handled will depend on the specifics of your control, its properties, and so on. The code is simple for the single property in the demonstration project, consisting of the single line shown in Listing 5.9.

Listing 5.9 The `ApplyChanges` Event Procedure

```
Private Sub PropertyPage_ApplyChanges()
SelectedControls(0). Caption = txtCaption.Text
End Sub
```

If we had permitted simultaneous changes to multiple selected controls, we could have used the following code:

```
Private Sub PropertyPage_ApplyChanges()
    'Declare a generic Object variable.
    Dim objControl As Variant
    'Loop through all selected controls.
    For Each objControl In SelectedControls
        objControl.Caption = txtCaption.Text
    Next
    End Sub
```

While we have created the property page for the `FancyCmdButton` control, we have yet to connect it to the control. Here are the required steps:

1. Double-click on `FancyCmdButton` in the Project window to open the designer.
2. In the Property list, scroll down to the `PropertyPages` property. The current setting of this property will be (none).

NOTES

3. Click on the button with the ellipsis (...) to display the Connect Property Pages dialog box (see Figure 5.11). The dialog lists the FCBGeneral page that we just designed, as well as three standard property pages that Visual Basic makes available to you.

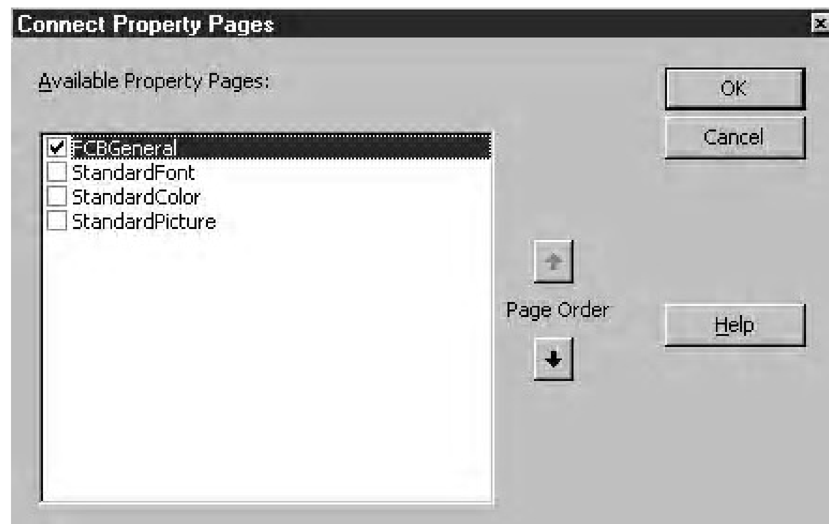


Fig. 5.11 Connecting a Property Page to the `FancyCmdButton` Control

4. Click on the FCBGeneral property page name to display a check mark in the box next to it, then click on OK.

Now that the property page is connected to the control, you can use it to set the control's properties—in this case, there is only one property. To try it out, you must first close the Property Page designer. Just like a control, a property page must be in the run mode to be available to its connected control. Then, double-click on Form1 in the Project box to display the form for the test program. Right-click on the `FancyCmdButton` control on the form and select Properties from the pop-up menu. The property page we designed will be displayed as shown in Figure 5.12.

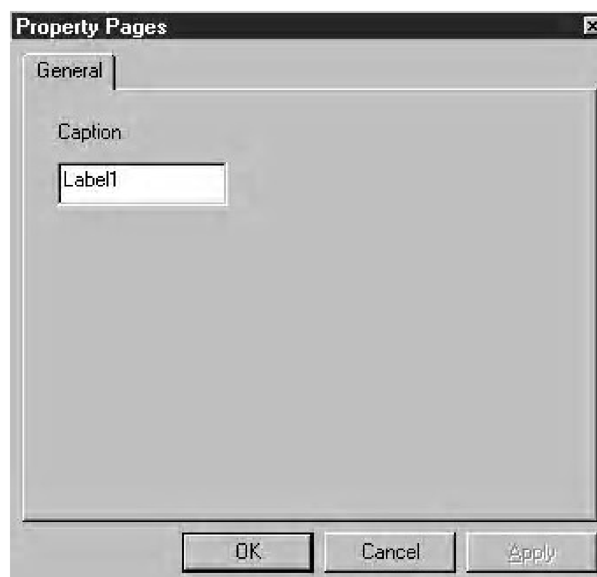


Fig. 5.12 Displaying the Property Page for the `FancyCmdButton` Control

Change the `Caption` property on the property page and, then click on either OK or Apply. You will see the new property reflected immediately on the control on the test form.

NOTES

Compiling ActiveX Control

As long as your ActiveX control is a part of a Visual Basic project, it can be used within that project—and only within it. To make it available to other applications, you must compile it into an OCX file. In this section, you will be taught how to compile the demonstration ActiveX control that we created and how to use the compiled version in your project. To compile the ActiveX control, follow the steps given below:

- Ensure that all the parts of the project group are in the design mode.
- In the Project window, click on AXCtrlDemo to make it an active project.
- Open the File menu and select Make AXCtrlDemo.ocx. Visual Basic displays the Make Project dialog box. If you want the OCX file in a different folder, select it here. You can accept the default name for the OCX file, which is the same as the project name (`AXCtrlDemo`). You can also assign a different name, such as FancyCmdButton, if you wish.
- Click on OK. Visual Basic will compile the project. No message is displayed upon completion, but if you look in the specified folder, you will find the OCX file.
- On the File menu, select Remove Project to remove the ActiveX control project from the project group. Visual Basic will display a warning message, because the control is referenced from another part of the project group—but that is okay.

Once you have compiled the ActiveX control into an OCX file and removed the ActiveX project from the project group, Visual Basic will automatically switch to using the compiled version in the test project. You will see that the icon for the ActiveX control is still displayed in the Visual Basic Toolbox. You can add other instances of control to the project's form, access its property page, and so on.

Distributing ActiveX Control in Other Projects

When you start a new Visual Basic project, you will not automatically have access to the ActiveX controls you have created. To add them to the Toolbox, you must select Components from the Project menu to display the Components dialog box (see Figure 5.13). Place a check mark next to the control or controls you want available in your project (controls you created are listed by the name you assigned). In the case of the demonstration control, it will be listed as Fancy Command Button. At that point, the control will be available to your project, just as any other ActiveX control.

You can distribute ActiveX control with the help of Visual Basic Package and Deployment Wizard.

NOTES

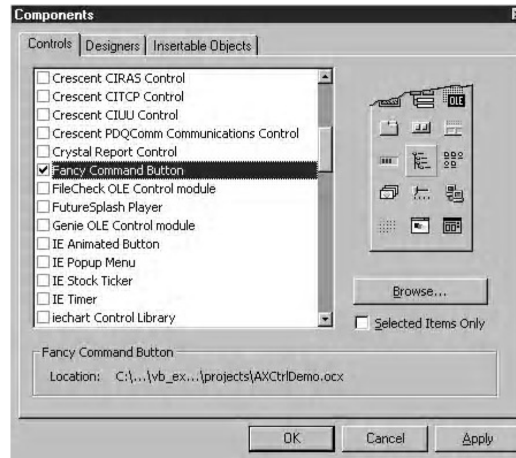


Fig. 5.13 Making an ActiveX Control Available in your Project with the Components Dialog Box

Built-In ActiveX Control

Visual Basic 6.0 provides us with a number of custom controls. Some of them are data bound controls and some are ActiveX controls. Since a majority of these controls happen to be ActiveX controls, these custom controls are also called ActiveX controls. An ActiveX control is an object that we place on a form to enable or enhance the user's interaction with an application. ActiveX controls have events and can be incorporated into other controls. These controls have a .ocx file name extension.

A custom control is an extension of the VB toolbox. A custom control file is a special form of a Dynamic Link Library (DLL) file that contains one or more types of controls. In addition, a custom control file can also contain exported functions that can be called directly from Visual Basic. When a custom control is added to the program, it becomes a part of the development and run-time environment and provides new functionality for the application. Visual Basic 6.0 supports ActiveX custom controls in addition to the VBX controls.

Custom controls can be added to and removed from the Toolbox using the Component dialog.

- From the Project menu, choose Component or right-click on the Toolbox and choose component from the pop-up menu. The component dialog opens (see Figure 5.14) for adding a controls to toolbox.

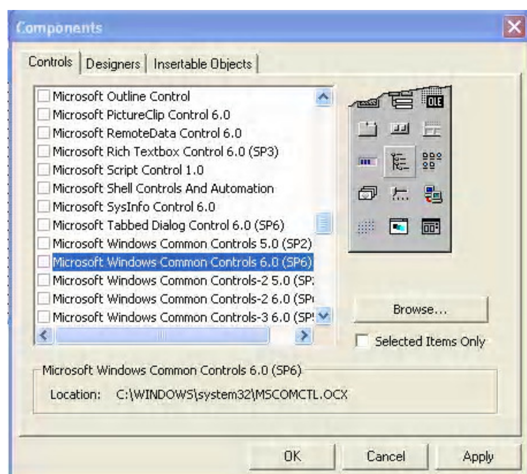


Fig. 5.14 Component Dialog Box For Adding a Controls to Toolbox

Put a check against the control, which you want to add to Toolbox.

- Click Close or click Apply and close the Component dialog when you finish adding the controls.

Now you can see the checked control icon in Toolbox.

NOTES



Visual Basic 6.0 offers us many built-in ActiveX controls of which a few are listed and briefly explained below.

Image List Control

Image list controls are invisible controls that serve one purpose, i.e., to hold images that are used by other controls. Usually, you add images to an image list control at design time, using the Insert Picture button in the control's property pages. You can also add images to an image list at run-time, using the Add method of its internal image collection, `ListImages`.



Image list controls: Invisible controls that serve one purpose to hold images that are used by other controls

To use the images in the image list, you usually associate the image list with a Windows common control (which has an `ImageList` property). For each item in the common control, such as a tab in a tab strip control, you can then specify either an index into the image lists' `ListImages` collection or an image's key value to associate that image with the item.

You can also reach the images in an image list with the `ListImages` collection's Picture property. For example, if you want to use an image list with a control that is not a Windows common control, such as a picture box, you can assign the first image in the image control to that picture box this way:

```
Picture1.Picture =
                    ImageList1.ListImages(1).Picture
```

The `ImageList` control tool appears in the Visual Basic toolbox as shown in Figure 5.15 at the bottom, on the right.

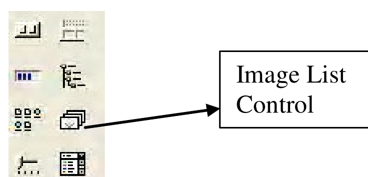


***Fig. 5.15** Toolbox Showing Image List Control*

NOTES

Adding an Image List to a Form

To work with many Windows common controls, you need to use image lists. Follow the steps given below to add an image list control to a program.

- Select the Project or Components menu item.
- Click the Controls tab in the Components dialog box that opens.
- Select the Windows Common Controls entry.
- Close the Components dialog box by clicking on OK.
- Double-click the Image List Control tool (see Figure 5.16 at the bottom, on the right) to add an image list control to a form. This control is invisible at run-time, so its size and location do not make much difference. Now that you have added an image list to a form, how do you add images to that image list? this is outlined in the next section.

Adding Images to Image Lists

To add images to an image list, you can use the image list's property pages at design time. Just right-click the image list and select the Properties item in the menu that opens. Next, click the Images tab in the property pages as shown in Figure 5.16.

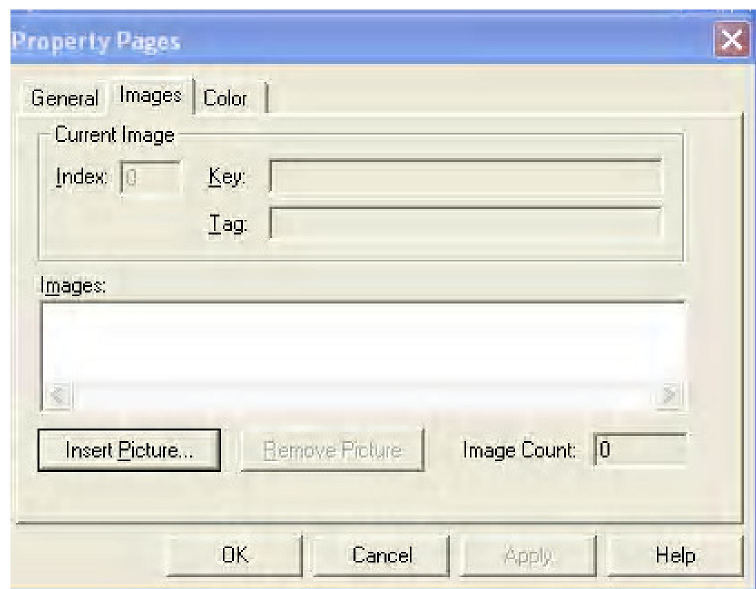


Fig. 5.16 Property Pages to Add Images to an Image List

To insert images into the image list control, just use the Insert Picture button; clicking that button lets you search for image files on disk. Each successive image gets a new `Index` value, starting from 1 and counting up. If you wish, you can also give each image a `Key` value (a unique text string identifier) by entering text in the box labelled `Key` when you add an image. When you are done adding images, close the property pages by clicking on OK. You can also add images to an image list using the `ListImages` collection's `Add` method at run-time like this, where we give the image the key 'tools':

```
    ImageList1.ListImages.Add , "tools",
LoadPicture("c:\tools.bmp")
```

NOTES

Tree View Control

If you have used the Windows Explorer, you are familiar with tree views. Tree views present data in a hierarchical way, such as the view of directories that appears in the tree view on the left of the Windows Explorer, as shown in Figure 5.17.

Trees are composed of cascading branches of *nodes*, and each node usually consists of an image (set with the `Image` property) and a label (set with the `Text` property). Images for the nodes are supplied by an image list control associated with the tree view control. A node can be expanded or collapsed, depending on whether or not the node has child nodes. At the topmost level are *root* nodes and each root node can have any number of child nodes. Each node in a tree is actually a programmable Node object, which belongs to the `Nodes` collection. As with other collections, each member of the collection has a unique `Index` and `Key` property that allows you to access the properties of the node.

Adding a Tree View to a Form

To add a tree view control to a form, follow the steps mentioned above for other controls.

Selecting Tree View Styles

There are many different styles for tree views—text nodes only, pictures and text nodes, showing or not showing the tree 'lines' that connect nodes, showing or not showing the plus and minus symbols to expand or collapse nodes, and so on. You set the tree view's style using its Style property. Given below are the possible values (we will stick to the default, style 7).

- tvwTextOnly—0
- tvwPictureText—1
- tvwPlusMinusText—2
- tvwPlusPictureText—3
- tvwTreelinesText—4
- tvwTreelinesPictureText—5
- tvwTreeLinesPlusMinusText—6
- tvwTreelinesPlusMinusPictureText—7 (the default)

Adding Nodes to a Tree View

You can add Node objects to a tree view by adding them to the `Nodes` collection. How does this work? For example, we will add a node, `Node1`, to a tree view, `TreeView1` (the tree view's `Style` property is set to `tvwTreelinesPlusMinusPictureText`, the default).

```
First, we declare the node:
Private Sub Form_Load()
Dim Node1 As Node
...
```

Next, we add the node to the tree view using the `Nodes` collection's `Add` method (see the next topic for more information on this method):

```
Private Sub Form_Load()
Dim Node1 As Node
Set Node1 = TreeView1.Nodes.Add
```

NOTES

```
...
```

Now we can refer to the node by name, `Node1`, as we set its text:

```
Private Sub Form_Load()
Dim Node1 As Node
Set Node1 = TreeView1.Nodes.Add
Node1.Text = "Node 1"
...
```

We can also refer to the node as a member of the `Nodes` collection as here, where we set the node's `Key` property:

```
Private Sub Form_Load()
Dim Node1 As Node
Set Node1 = TreeView1.Nodes.Add
Node1.Text = "Node 1"
TreeView1.Nodes(1).Key = "Node 1"
End Sub
```

How does this look when you run it? You can see that the result in Figure 5.18 is not very spectacular with just one node. You can add other nodes by duplicating the preceding code and naming the new nodes `Node2`, `Node3`, and so on, but they will all appear at the same level.

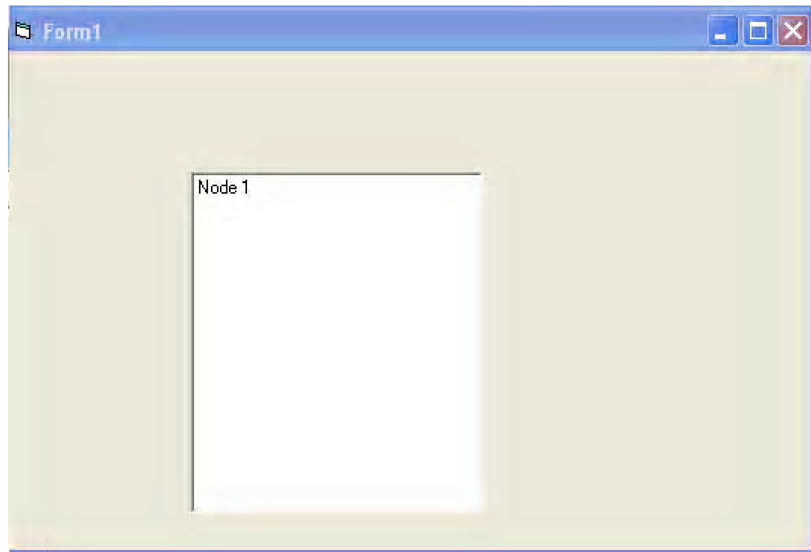


Fig. 5.18 Result of a Program

Uses of Tree View Control

- To create an organization tree that can be manipulated by the user.
- To create a hierarchy that displays at least two or more level of a database.

`ListView` Control

The `ListView` control displays lists of items. You can see a list view on the right in the Windows Explorer in Figure 5.19. Here, the list view is displaying a list of files. Each item in a `ListView` control is itself a `ListItem` object and can have both text and an image associated with it. The `ListItem` objects are stored in the list view's `ListItems` collection.

List views can display data in four different view modes:

- ***Icon mode***—Can be manipulated with the mouse, allowing the user to drag and drop and rearrange objects.
- ***SmallIcon mode***—Allows more `ListItem` objects to be viewed. Like the Icon view mode, objects can be rearranged by the user.
- ***List mode***— Presents a sorted view of the `ListItem` objects.
- ***Report mode***—Presents a sorted view with sub-items, allowing extra information to be displayed.

Uses of ListView Control

- To display the result of a query on a database.
- To display all the records in a database table.
- In association with a `TreeView` control, to give users an expanded view of a `TreeView` control node.

Tab Strip Control

A tab strip control presents the user with a row (or rows) of tabs that acts like the divider in a notebook or the labels on a group of file folders. Like an increasing number of other controls (such as cool bars and tree views), tab strips represent one of Microsoft's attempts to compact data into less and less of the screen (because there's getting to be more and more data). Using tab strips, the user can click on a tab and see a whole new panel of data, like opening a file folder.

Uses of Tab Strip Control

- To create a tabbed dialog that sets various text attributes for a `RichTextBox` control
- To create a tabbed dialog that sets preferences for an application

Status Bar Control

Status bars appear at the bottom of windows and usually hold several panels in which you can display text. The status bar is there to give feedback to the user on program operation as well as other items like the time of the day or key states (such as, the Caps Lock or the Ins key). Although status bars usually display text in panels, there is a simple status bar style that makes the status bar function as one long panel, as we will see. Status bars are built around the `Panels` collection, which holds the panels in the status bar. Upto 16 Panel objects can be contained in the collection. Each object can display an image and text. You can change the text, images or widths of any Panel object, using its `Text`, `Picture` and `Width` properties. To add Panel objects at design time, right-click the status bar and click Properties to display the Property Pages dialog box. A status bar can be placed at the top, bottom or sides of an application. Optionally, the control can be float within the application's client area.

Uses of Status Bar Control

- To inform the user of a database table's metrics, such as number of records, and the present position in the database.
- To give the user information about a `RichTextBox` control text and font status.
- To give the user status about key states (such as the Caps Lock or the Number Lock).

NOTES

Progress Bars

Progress bars give the user some visual feedback on what is happening during a time-consuming operation. They present the user with a colour bar that grows in the control to show how the operation is proceeding, usually from 0 to 100 per cent. You can use a progress bar when an operation takes some time to finish. The progress bar's Value property (not available at design time) determines how much of the control has been filled. The Min and Max properties set the limits of the control. You add the Progress Bar Control tool to the toolbox by following the same steps to add the toolbar tool, because the progress bar control is also a part of the Microsoft Windows common controls. The Progress Bar Control tool is the thirteenth tool down on the left.

Uses of Progress Bar Control

- To inform the user of progress as a file transfer occurs over a network.
- To reflect the state of a process that takes more than a few seconds.
- To inform the user about the progress of a complex algorithm.

5.3 CREATING AN ACTIVEX CONTROL TO ACTIVATE A WEB PAGE

In 1996, Microsoft launched its ActiveX technology which provided a simple technique to programmers for offering additional facilities to end-user applications. Basically, ActiveX controls are used to provide browser enhancements, such as advanced multimedia capabilities and can be incorporated into software, such as the Microsoft Office family of applications. The disadvantage of using ActiveX controls is that it enhances security risk because the programmers can hide code within an apparently safe control to attack a user's computer.

In fact, ActiveX technology is exclusively developed by Microsoft. In it with an ActiveX-enabled browser, i.e., Internet Explorer only ActiveX controls can be safely downloaded as part of a Web document for adding functionality to the Web browser. Especially, ActiveX enables faultless viewing of all sorts of Windows files in combination with other technologies, such as Java and scripting languages that helps in the development of complex Web applications.

An ActiveX component is basically a user defined executable control that can be used to create interfaces and applications like other controls, such as `TextBox` and `CommandButton`. ActiveX components allow programmers to reuse these user defined controls to perform manipulation on various data types in their applications. ActiveX components are saved as a file with .exe, .dll or .ocx extensions.

Reusing the built-in components aids the programmer for early software development. ActiveX components were previously known as OLE controls. ActiveX components allow the programmers to add functionality to the Web pages. You can easily design ActiveX components by using the standard controls available in the Visual Studio Toolbox. You can directly test and debug the ActiveX component code during the designing of the ActiveX control. You can bind the ActiveX components with the individual controls on a form to control the flow of information that needs to be performed on the form.

You can integrate all the controls that collectively form the interface. The integration of all the controls is used while creating the ActiveX component as it allows the programmer to assign property, events and methods collectively. After designing and coding the ActiveX component, you may compile the ActiveX component as an .ocx file. The .ocx file formed after compiling the ActiveX component is a binary executable file that can be integrated with the application during the development phase. The .ocx file can also be integrated with various development tools, such as Microsoft Visual Basic and Microsoft Visual C++.

You can integrate your ActiveX component with Web based applications as well. Use of ActiveX components in Web based applications has proved to be a boon for programmers. ActiveX components allow the programmers to design controls by using built-in controls. ActiveX components built using Visual basic can be easily integrated with Microsoft Office.

Creating User Defined ActiveX Controls

VB provides various methods for the ActiveX control. While working with ActiveX control, you can implement these methods in your project to display information regarding the working of ActiveX control. These methods execute in a specific sequence when you run your project. Following ordered list shows the various methods and their sequence of execution:

- `UserControl_Initialize`
- `UserControl_ReadProperties`
- `UserControl_Resize`
- `UserControl_Show`
- `UserControl_Terminate`

ActiveX controls provide the programmer with the flexibility to reuse the integrated component. The following steps create a new user defined ActiveX control and show how you can implement the above methods in an ActiveX control:

1. Open a new project in Visual Basic and select the ActiveX Control option from the New tab page. Click the Open button to create a new ActiveX Control object.

Figure 5.19 shows the `UserControl1` window.

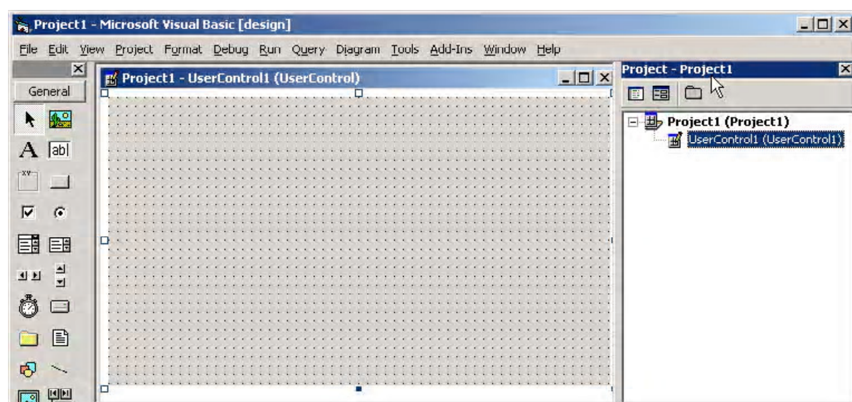


Fig. 5.19 The UserControl1 Window

NOTES

2. Change the `BackColor` property of the form in the `UserControl1` window to `ToolTip` to change the background color. Figure 5.20 shows the `UserControl1` window with changed background color.

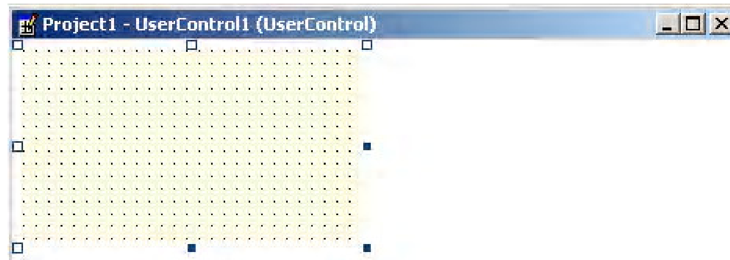


Fig. 5.20 Displaying `UserControl1` Window with Changed form Color

3. Close the `UserControl1` window to enable the `UserControl1` icon in the Toolbox.
4. Right click the Project1 in the Project explorer window and select the Form option to add a form to the project.
5. Drag and drop the `UserControl` on the Form1 form, as shown in the following figure. Figure 5.21 shows the Form1 window with the created `UserControl`.

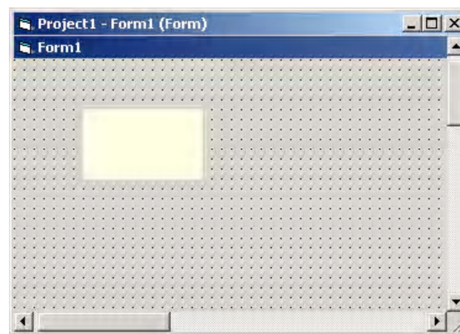


Fig. 5.21 Displaying Form1 Window with the Created `UserControl`

6. Now, to associate code with the `UserControl`, right click the `UserControl1` node in the Project1 explorer window and select View Code option to open the Code window.
7. Enter the following lines of code in the Code window:

```
Private Sub UserControl_Initialize()
MsgBox "Initializing User Control"
End Sub
Private Sub UserControl_ReadProperties(xwr As _
PropertyBag)
MsgBox "Checking ReadProperties"
End Sub
Private Sub UserControl_Resize()
MsgBox "Resizing " & ScaleWidth & ", " & ScaleHeight
End Sub
Private Sub UserControl_Show()
MsgBox "Display User Control"
```

```
End Sub
Private Sub UserControl_Terminate()
MsgBox "Terminating User Control"
End Sub
```

Figure 5.22 shows the Code window for creating user defined ActiveX control.

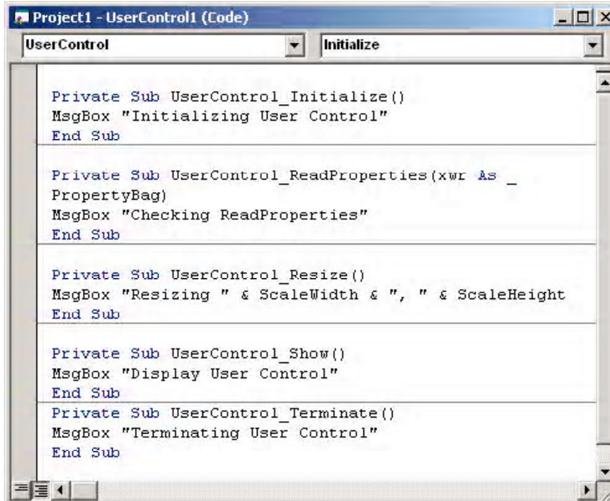


Fig. 5.22 *Displaying Code Window for Creating ActiveX Control*

8. Close the `UserControl1` and Form1 window and press F5 to compile and execute the code for ActiveX control.

When the program executes, first of all `UserControl_Initialize` method executes and displays the following figure. Figure 5.23 shows the Project1 dialog box for `UserControl_Initialize` method.



Fig. 5.23 *Displaying Output for `UserControl_Initialize` Method*

After executing the `UserControl_Initialize` method, `UserControl_Resize` method executes. Figure 5.24 shows the output for `UserControl_Resize` method.

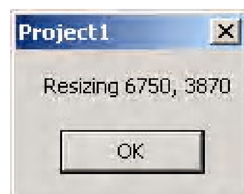


Fig. 5.24 *Displaying the Output for `UserControl_Resize` Method*

Similarly, `UserControl_ReadProperties` and `UserControl_Show` methods execute.

Note: The `UserControl_Terminate` method executes only when a user closes the Form1 form during the execution process.

NOTES

NOTES

Let us consider another example where you need to use an inbuilt Microsoft ADO Data Control 6.0 ActiveX control to retrieve data from a database. For this you need to perform the following sequence of steps:

1. Open the UserControl1 window.
2. Drag and drop the label on the form and change the `Caption` property to `Northwind Employee` names. Select the `Font` property from the Properties Window to display the Font dialog box. Figure 5.25 shows the Font dialog box.

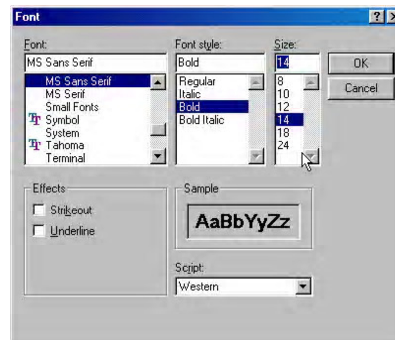


Fig. 5.25 *The Font Dialog Box*

3. Select the MS Sans Serif option from the Font `ComboBox`, Bold option from the Font style `ComboBox` and 14 from the Size `ComboBox`. Click the OK button to confirm the settings.
4. Drag and drop three labels and change the `Caption` property to First Name, Last Name and Birth Date correspondingly.
5. Select the `Font` property using the ellipses button in the Properties window to display the Font dialog box. Select the System option from the Font combo box and the Bold Italic option from the Font style combo box.
6. Drag and drop three `TextBox` from the Toolbox window.
7. Drag and drop one command button and change the `Caption` property to Click Me.
8. Right click on the Toolbox window and select the Components option from the shortcut menu. The components dialog box is displayed.

Figure 5.26 shows the Component dialog box.

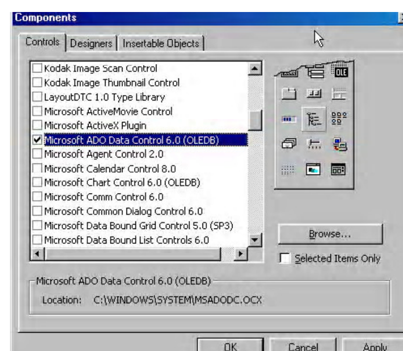


Fig. 5.26 *The Component Dialog Box*

9. Select the Microsoft ADO Data Control 6.0 (OLEDB) option from the Control tab page. Click the OK button to confirm the settings.

Note: You can see that the ADODC control is added to the Toolbox window.

10. Double click the ADODC to display it on the form and change the `Caption` property to **Navigate Records**.
11. Place all the controls on the form.

Figure 5.27 shows the placement of controls on the form.

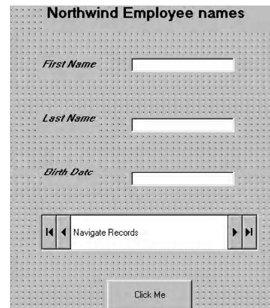


Fig. 5.27 *The Placement of Controls on the Form*

12. Right click the ADODC and select the `Properties` option from the shortcut menu. The Property Pages dialog box is displayed. Figure 5.28 shows the Property Pages dialog box.

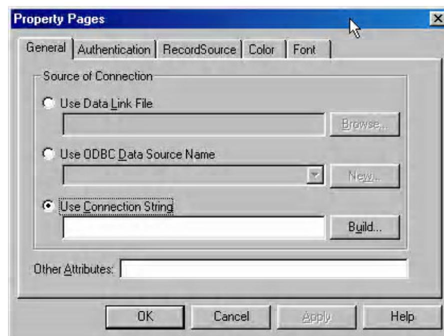


Fig. 5.28 *The Property Pages Dialog Box*

13. Click the Build button to display the Data Link Properties dialog box.

Figure 5.29 shows the Data Link Properties dialog box.

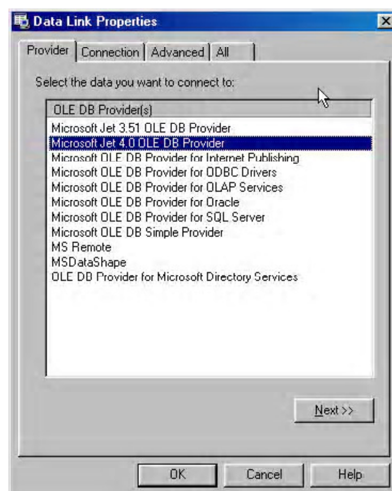


Fig. 5.29 *The Data Link Properties Dialog Box*

NOTES

NOTES

14. Select the Microsoft Jet 4.0 OLE DB Provider option from the OLE DB Provider(s) section.
15. Click the Connection tab to display the Connection tab page. Specify the path for the Access database by using the Browse button. Figure 5.30 shows the Connection tab page.

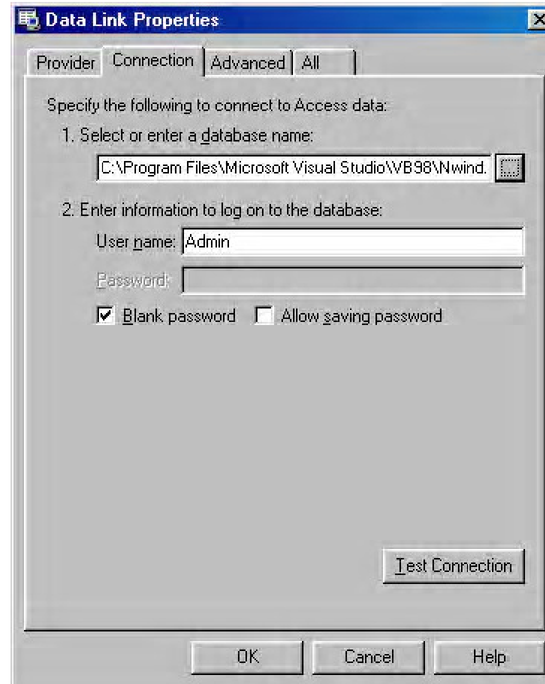


Fig. 5.30 *The Connection Tab Page*

16. Click the OK button to close the Data Link Properties dialog box.
17. Click the Test Connection button to verify that the connection to the access database files has been established. The Microsoft Data Link dialog box will display the message that the connection succeeds. Figure 5.31 shows the Microsoft Data Link dialog box.

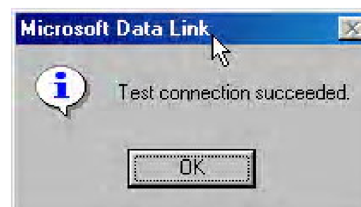


Fig. 5.31 *The Microsoft Data Link Dialog Box*

18. Click the OK button on the Microsoft Data Link dialog box.

Note: You can see that the `Use Connection String` option on the Property Pages dialog box has now been provided with the `Provider=Microsoft.Jet. OLEDB.4.0`; Persist `Security Info=False` value.

19. Click the `RecordSource` tab on the Property Pages dialog box.
20. Select the `2 – adCmdTable` option from the `Command Type` drop-down list and Employees option from the Table or Stored Procedure Name drop-down list. Figure 5.32 shows the `RecordSource` tab page.

NOTES

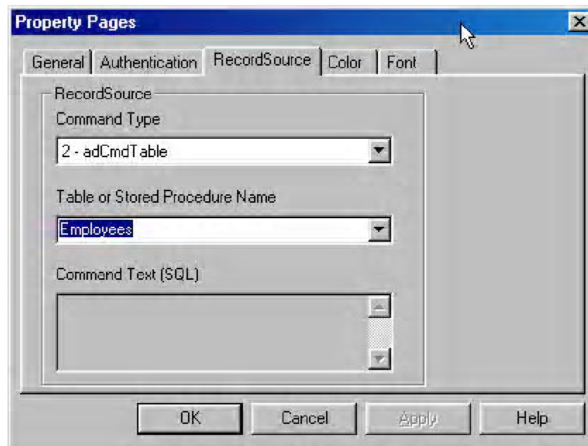


Fig. 5.32 The `RecordSource` *Tab Page*

21. Click the OK button to close the Property Pages dialog box.
22. Select all the three `TextBoxes` on the interface and select the adodc option from the `DataSource` drop-down list.
23. Select the Text1 `TextBox` and set the `DataField` property to First Name.
24. Select the Text2 `TextBox` and set the `DataField` property to Last Name.
25. Select the Text3 `TextBox` and set the `DataField` property to Birth Date.

Press F5 to execute the ActiveX control. The Internet Explorer Window opens to display the output. Figure 5.33 shows the ActiveX control.

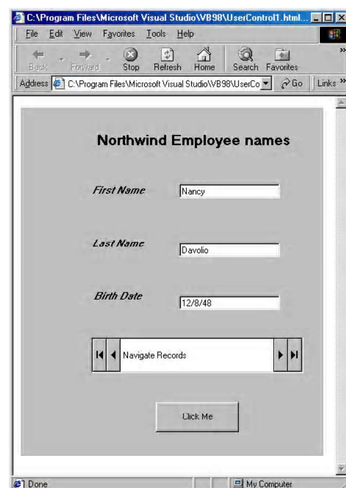


Fig. 5.33 The User Defined ActiveX Control

Benefits of ActiveX Controls

The concept of ActiveX was developed for very simple reasons that add to its various benefits:

- ActiveX components allow the control to import data from one application to another. It also allows you to make changes in the imported data such that the data in the parent application can also be updated.

NOTES

- ActiveX components allow the programmer to place different types of data or 'objects' in one document.
- It allows applications to exchange data between various applications. It also allows one application to send commands to another application.
- It also allows various objects from different applications to be visible in a single document. The imported object gets embedded in the host application document. When a user clicks on the object, the parent application of that object gets invoked to present the called object.
- ActiveX prevents the conversion of data while integrating one application into another.

Adding Property to the Control

After creating the ActiveX control, you need to add property, events and methods to the control to determine the functionality of the user defined control. While assigning a property to the control, you may also want to map the standard property to your control. You can define the standard property to your control in following two ways:

- Using the ActiveX Interface Control Wizard.
- Using the Add Property Dialog Box.

Using the ActiveX Interface Control Wizard

You can launch the ActiveX Control Interface Wizard by using the Add –Ins menu on the Visual Basic window.

1. Select the ActiveX Control Interface Wizard option from the Add-Ins menu. Figure 5.34 shows the ActiveX Control Interface Wizard screen.

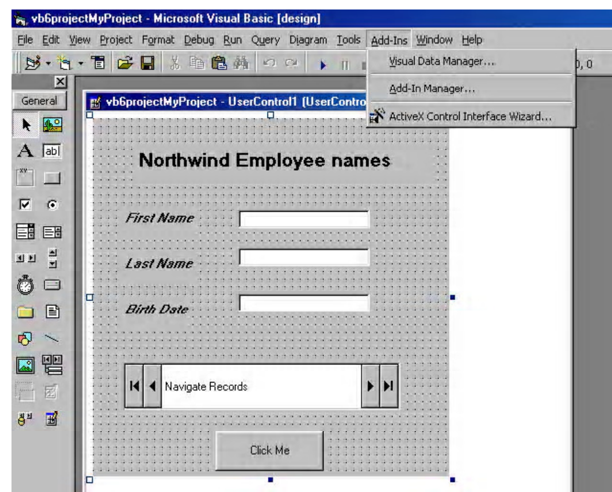


Fig. 5.34 *The ActiveX Control Interface Wizard Screen*

The introductory screen of the ActiveX Control Interface wizard is displayed that provides information about how this wizard allows the programmer to assign property to the control.

Note: It is good practice to lay all the components on the interface before assigning them property using the ActiveX Control Interface Wizard.

Figure 5.35 shows the ActiveX Control Interface Wizard – Introduction screen.

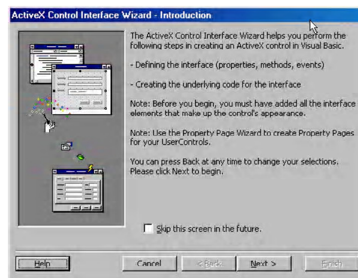


Fig. 5.35 The ActiveX Control Interface Wizard – Introduction Screen

2. Click the Next button to display the ActiveX Control Interface Wizard – Select Interface Members screen. This screen allows you to select single or multiple custom properties defined for custom controls. To select custom property, highlight the property in the Available names section and click the right arrow button to add the property to the Selected names section.

   Figure 5.36 shows the selected `BackColor` and `DataSource` properties.



Fig. 5.36 The Selected Properties

3. Click the Next button to display the ActiveX Control Interface Wizard – Create Custom Interface Members screen. This screen allows you to define some of the custom members for the control. You can create new members by clicking the New button. If there exists some custom members for the control, the Edit and Delete buttons will appear enabled to modify and delete the existing custom members.

   Figure 5.37 shows the ActiveX Control Interface Wizard – Create Custom Interface Members screen.



Fig. 5.37 The ActiveX Control Interface Wizard – Create Custom Interface Members Screen

NOTES

4. Click the Next button to display the ActiveX Control Interface Wizard – Set Mapping screen. This screen allows you to map the custom property with various controls defined in the interface. After selecting the interface control from the Control drop-down `ListBox`, you need to determine the control member to which the custom property needs to be mapped.

   Figure 5.38 shows the ActiveX Control Interface Wizard – Set Mapping screen.



Fig. 5.38 *The ActiveX Control Interface Wizard – Set Mapping Screen*

In the above figure, the `BackColor` custom property is mapped to the `BackColor` member property of the ADODC1 control

1. Click the Next button to display the ActiveX Control Interface Wizard – Finished! screen. This screen displays the message that the information required to set the property of the ActiveX control has been collected.

   Figure 5.39 shows the ActiveX Control Interface Wizard – Finished! screen.

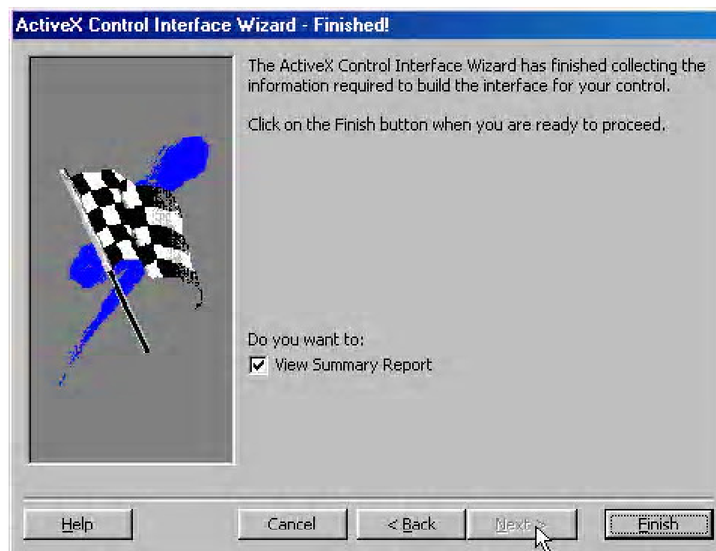


Fig. 5.39 *The ActiveX Control Interface Wizard – Finished! Screen*

2. Click the Finish button to close the wizard. Before closing, the ActiveX Control Interface Wizard Summary screen is displayed. This screen summarizes the information required to test and debug the ActiveX Control.

Figure 5.40 shows the ActiveX Control Interface Wizard Summary screen.

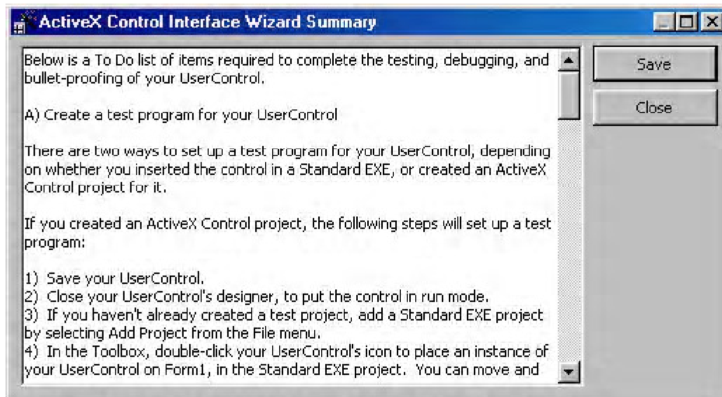


Fig. 5.40 The ActiveX Control Interface Wizard Summary Screen

3. Click the Close button to close the wizard. Open the Code window for the interface. You can see that multiple lines of code have been generated in the existing code for the properties which you have defined by using the ActiveX Control Interface Wizard.

   Figure 5.41 shows the Code window for the interface.

```
Private Sub Command1_Click()
MsgBox ("You are using the user control")
End Sub

'WARNING! DO NOT REMOVE OR MODIFY THE FOLLOWING COMMENTED LINES!
'MappingInfo=Adodc1,Adodc1,-1,BackColor
Public Property Get BackColor() As OLE_COLOR
    BackColor = Adodc1.BackColor
End Property

Public Property Let BackColor(ByVal New_BackColor As OLE_COLOR)
    Adodc1.BackColor() = New_BackColor
    PropertyChanged "BackColor"
End Property

'WARNING! DO NOT REMOVE OR MODIFY THE FOLLOWING COMMENTED LINES!
'MemberInfo=32,0,0,0
Public Property Get DataSource() As DataSource
    Set DataSource = m_DataSource
End Property

Public Property Set DataSource(ByVal New_DataSource As DataSource)
    Set m_DataSource = New_DataSource
    PropertyChanged "DataSource"
End Property

'Load property values from storage
Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
    Adodc1.BackColor = PropBag.ReadProperty("BackColor", &H80000005)
    Set m_DataSource = PropBag.ReadProperty("DataSource", Nothing)
End Sub

'Write property values to storage
Private Sub UserControl_WriteProperties(PropBag As PropertyBag)
    Call PropBag.WriteProperty("BackColor", Adodc1.BackColor, &H80000005)
    Call PropBag.WriteProperty("DataSource", m_DataSource, Nothing)
End Sub
```

Fig. 5.41 The Code Window for the Interface

Using the Add Property Dialog Box

You can also add the custom property to the controls in an interface by using the Add Property dialog box. The following steps can be performed for this:

1. To add property to the custom control, select the Add Procedure option from the Tools menu. This will launch the Add Procedure dialog box.

   Figure 5.42 shows the Add Procedure dialog box.

NOTES

NOTES

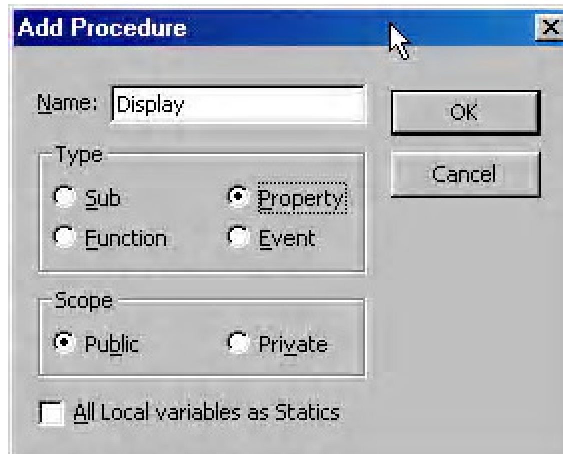


Fig. 5.42 *The Add Procedure Dialog Box*

2. Enter the name of the procedure in the Name `TextBox`.
3. Select the type of the procedure from the Type section as `Property` to set the property for the control.
4. Select the scope of the procedure as Public or Private from the Scope section.
5. Click the OK button to create the procedure.

Note: You can verify that the following lines of code have been added to the Code window;

```
Public Property Get Display() As Variant
End Property
```

The Add Procedure dialog box simply creates the skeleton for the procedure code. Now, you can define the functionality of the control in the code as per the requirements of the application.

Creating ActiveX DLL Component

You need to create an ActiveX DLL component to create binary executables that can be integrated with other applications. ActiveX DLL components are based on the in-process architecture that executes on the same memory space in which the executable files are executed. Let us create a simple ActiveX DLL component which calls the Windows DLL file to open the notepad. As the ActiveX DLL components are not selfexecutable, you need to create an ActiveX exe application that calls the `openingnotepad` function of the ActiveX DLL file.

To create an ActiveX DLL component, you need to perform the following steps:

1. Launch the Visual Basic 6 design environment and select the ActiveX DLL option from the New Project dialog box. The Code window is displayed.
2. Select the Add Procedure option from the Tools menu to create a new function.
3. Name the function as `openingnotepad`. You will see the following lines of code on the Code window as:

```
Public Function openingnotepad()
End Function
```

4. Write the following lines of code for the function:

```
Shell  ("notepad.exe")
```

Figure 5.43 shows the code for the DLL file.

NOTES

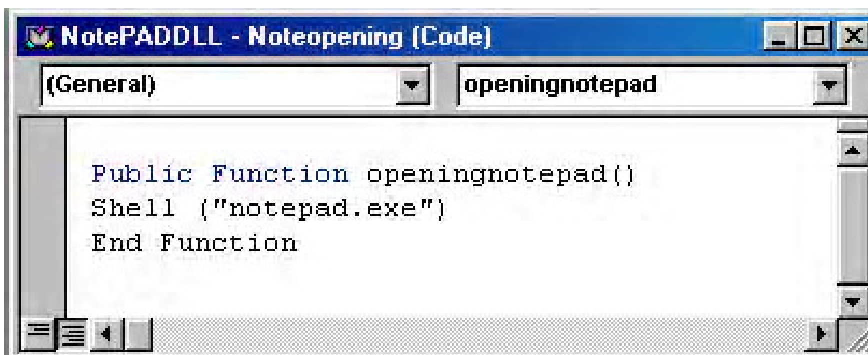


Fig. 5.43 The Code for the DLL File

5. Save the class file with the name noteopening and the project with the name notepad.dll.
6. Compile the file and save it in the folder where various other Visual Basic projects are saved.
7. Select the New option from the File menu to launch a new Visual Basic project.
8. Select the Standard EXE option from the New Project dialog box to create a container file for the DLL component.
9. Select the References option from the Project menu to display the References dialog box.

Figure 5.44 shows the References dialog box.

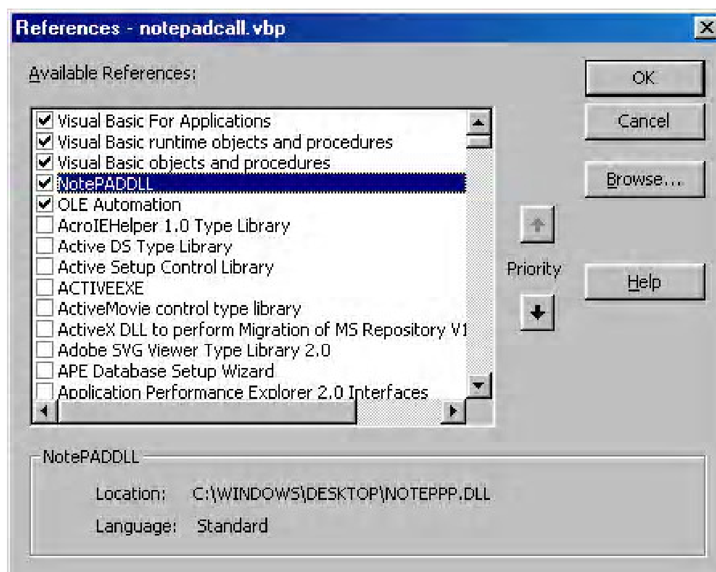


Fig. 5.44 The References Dialog Box

10. Click the Browse button to navigate to the DLL file to add its reference to the exe file. Click the OK button to close the References dialog box.

NOTES

11. Drag and drop one command button on the form and change the caption to Click Me.
12. Double click the `commandButton` to display the Code window and write the following lines of code:

```
Dim a As New Noteopening
a.openingnotepad
```

Note: Make sure you write the above code for the `Click` event of the `commandButton`.

13. Save the project and form with the names, notepad.exe and notepad.form, correspondingly.
14. Execute the form by pressing the F5 key from the keyboard.
15. Click the Click Me button. You will notice that the notepad file is open for you on the desktop.

Built-In ActiveX Controls

VB also provides various built-in ActiveX controls, such as Animation and Communication which you can directly use in your programming application. The Table 5.4 lists the various commonly used ActiveX controls:

Table 5.4 *Various Built-In ActiveX Controls and Description*

ActiveX Control	Description
`Animation`	Displays only the silent AVI files. An AVI file contains a series of bitmap image frames. An AVI file can also contain sound files along with the bitmap frames. However, you cannot use these files with the `Animation` control.
`Communication`	Provides an interface that allows you to connect to a serial port and other communication devices, such as modem to communicate.
`DateTimePicker`	Displays the date and time and also provides an interface that enables you to change the data and time information.
`FlatScrollBar`	Is just like a standard window `ScrollBar`, which enables you to scroll vertically and horizontally.
`ImageCombo` Control	Is like a `ComboBox` with the ability to include images along with each item in the ImageCombo list.
Internet Transfer	Implements the HyperText Transfer protocol (HTTP) and File Transfer protocol (FTP). The Internet Transfer control allows you to connect to any Web site that uses either HTTP of FTP to retrieve the site information or data files.
`ListView`	Displays data as `ListItem` objects. A `ListItem` object is an optional icon associated with a label of the object.
Messaging Application Program Interface (MAPI)	Enables you to create a mail enabled VB application.
`RichTextBox`	Is like a `TextBox` that enables a user to enter or edit text. `RichTextBox` also provides various formatting features, such as creating subscripts, creating superscripts and changing the text color.
`StatusBar`	Is a frame, which can contain more than one panel. The `StatusBar` control informs a user about the status of an application.
`TreeView`	Is used to display data in a hierarchical order, such as files and directories on a disk.

Registering and Unregistering Components

You need to perform the following sequence of steps to register an ActiveX component (.ocx file). To register an ActiveX control:

NOTES

1. Select Start→Programs→MS DOS Prompt to display the command prompt window.
2. Browse the `Regsvr32` file on the command prompt and enter the following command on the command prompt window:

   Regsvr32.exe Northwind.ocx

   In the above command, Regsvr32 .exe is a VB tool, which is used to register the ActiveX component and Northwind.ocx is the name of the ActiveX component that you want to register.
3. Press the Enter key to run the command. A dialog box appears, which specifies that the Northwind.ocx component has been registered.

   Figure 5.45 shows the RegSvr32 dialog box.



Fig. 5.45 *The RegSvr32 Dialog Box*

You can also unregister a registered component. To unregister the registered Northwind.ocx component, enter the following command at the command prompt:

```
Regsvr32.exe \u Northwind.ocx
```

5.4 VDOLIVE TECHNOLOGY

VDOLive is a quite latest technology developed for supporting the broadcasting of both audio and video over the Internet or intranet. Though it resembles the audio broadcasts using RealAudio, it specifically adds a video signal to the stream which is the unique feature to it.

Conventionally, the user need to download the entire file and display it using any other external program if a Web page contains a video. To enable downloading and its automatic viewing the plug-ins and helper application of the browser are required and downloading files of bigger size can be a disappointing task if you have low Internet speed.

Scaleable video, algorithm for audio compression and the protocol that is used for the transmission of files over the Internet forms the basis of this technology. It facilitates the viewing and downloading the videos simultaneously. The quality of the picture is directly proportional to the available bandwidth of the Internet link. The video file starts downloading at the same time the player displays the embedded video which is resided in the memory buffer. There is no need to download the entire video to watch but one can watch it while downloading it from the server with VDOLive technology.

NOTES

Working Mechanisms of VDOLive

VDOLive is based on the client-server architecture where the client is VDOLive player and the server is a VDOLive server. The client enables users to receive and view video clips, whereas the server transmits video over the Internet to users. Figure 5.46 summarizes the function of the typical VDOLive system.

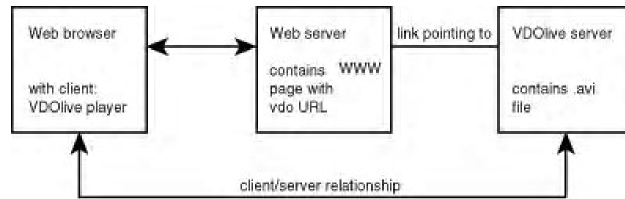


***Fig. 5.46** Connectivity between VDOLive player and VDOLive Server*

One must install a plug-in or helper application as standard browsers does not supports and display real time video and audio. VDOLive player is the plug-in that is used to view the VDOLive video clips and is available for the Windows and Macintosh platform which works with most of the Web browsers. This special purpose program runs on the client machine, interprets the video stream and displays it on a screen.

VDOLive server runs in the background and listens to the request made to the default TCP port. It then reads the file of VDO format from local file system and transmits it over the Internet to a client using the UDP protocol.

5.5 CREATING NETSCAPE NAVIGATOR PLUG-INS

Netscape Navigator provided a basic browser program for the Web. When Netscape first developed their Web browser, they decided they could never provide everyone who used it with every possible viewer for all files and they chose to open up their platform for outside developers.

Additional programs that make Navigator even more powerful are called plug-ins, and they provide Netscape Navigator with the ability to view non-HTML documents. They allow the user to extend the functions in Netscape Navigator to display multimedia, word and desktop publishing files and also features like three dimensional viewing, and real time chatting over the Internet.

Netscape Plug-in Application Programming Interface (NPAPI) is a cross-platform plugin architecture used by many web browsers. It was first developed for the Netscape family of browsers starting with Netscape Navigator 2.0 but was subsequently implemented by many other browsers as well. It also supports scripting, printing, full screen plugins, windowless plugins and content streaming.



Navigator Plug-Ins: Plug-ins are similar in functionality to Java. In contrast to Java, Navigator plug-ins are binary code modules written and compiled for each host operating system

Navigator Plug-Ins

Plug-ins are similar in functionality to Java. In contrast to Java, Navigator plug-ins are binary code modules written and compiled for each host operating system. A question may arise, why write a plug-in instead of a Java applet? The answer is speed which is provided by plug-ins. Navigator plug-ins are well suited for handling high bandwidth data transfers needed for demanding data types such as audio and video over the Internet.

Navigator plug-ins are file format driven. For example: A plug-in called FIGleaf Inline from Carberry Technologies. This plug-in supports image files formats such as GIF, JPEG, PNG, TIFF etc.

Architecture

Netscape Navigator's plug-in architecture is based on dynamically loaded code modules. These modules are present in a subfolder or directory called PLUGINS. Each module has a resource that determines which MIME type it can handle. When Navigator finds this MIME type embedded in a Web page through HTML or as a single file, it loads the appropriate code module.

When the browser scans its plugin directories and finds a new library, the browser will attempt to see if the library is a plugin. The following things must happen in order for the library to be considered as a plugin:

- The browser must be able to tell the plugin the addresses of the browser-side NAPI functions
- The plugin must be able to tell the browser the addresses of the plugin-side NAPI functions
- The browser must be able to determine what MIME types the plugin handles
- The browser must be able to determine the plugin's shutdown routine

MIME Types

A plug-in can support more than one MIME type. This feature can be really handy for plug-ins that support more than one file format. For example in Netscape's LiveAudio plug-in, the plug-in currently supports seven different MIME types. Let us see the MIME types, descriptions, and suffixes supported by the LiveAudio plug-in:

- audio/basic
- audio/x-aiff
- audio/aiff
- audio/x-wav
- audio/wav
- audio/x-midi
- audio/midi

Streaming With Netscape Navigator

Navigator 3.*x* enables you to stream data using Navigator plug-in. This has to be done by creating a new stream with `NPN_NewStream`, writing to it with `NPN_Write`, and destroying it with `NPN_DestroyStream`.

Netscape was discontinued and support for all Netscape browsers and client products was terminated on March 1, 2008.

5.6 PULLING WEB INFORMATION

The advent of the Internet and WWW drastically changed the life of people for better. It has altogether altered the way people access information. With the help of WWW, a myriad of Websites dealing with diverse subjects have come into existence. Millions of computers with billions of Web pages are hooked to the Internet and are ready to

NOTES

Check Your Progress

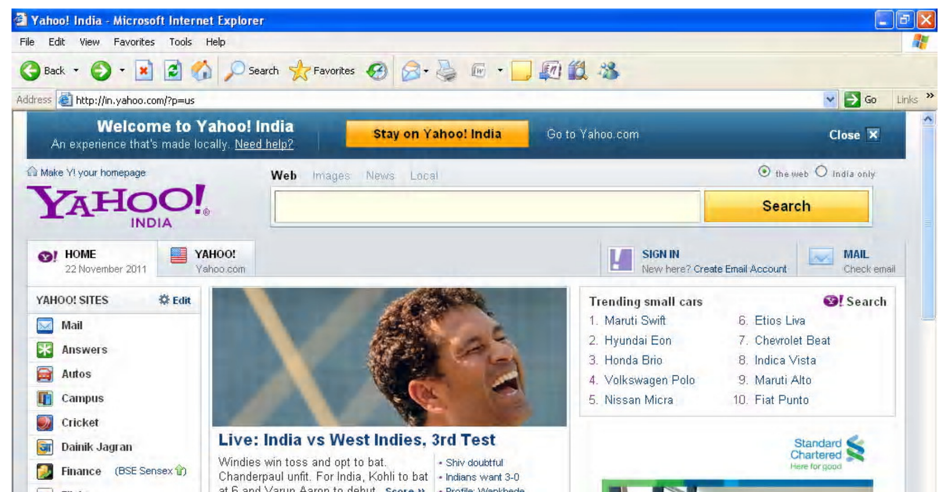
1. What is an ActiveX DLL component?
2. What is an ActiveX EXE?
3. State the advantages of DLLs.
4. What are the advantages of ActiveX control?
5. What are the disadvantages of ActiveX control?
6. How can you test an ActiveX control that you are developing?
7. What happens when an ActiveX control is placed on a Visual Basic form during program design?
8. What are the uses of Property pages?

NOTES

provide information and knowledge pertaining to any subject or topic of your choice. To extract information from the Internet, we need to explore the Internet. Surfing the Internet entails searching for two types of materials, namely the textual and the non-textual materials. HyperText Transfer Protocol (HTTP) facilitates exchange of documents between two or more computers that are connected to the Internet. It is the Web browser which has the ability to provide requisite documents in the form of Web pages. As an Internet is a repertoire of information, one needs to be adept in digging out the relevant information from such a vast ocean of Web pages. Basically, there are two popular methods for surfing the Internet. In the first method, you know the Web address of the particular Website in which the pertinent information can be searched for. In this case, you just need to key the Web address in the form of URL in the address bar of your Web browser. It enables you to see the homepage of the desired Website which allows navigation of different Web pages contained in that Website. The second method necessitates the use of search engines, that is, the software systems which enable the users to search for information on the WWW using specific keywords. Internet users key some keywords in the space provided in the search engine page. Obtaining the desired information is contingent upon the keywords entered to a great extent. Surfing the Internet with the help of search engines has become an integral part of our life. Surfing the Internet also facilitates access to chat rooms where online discussions or chatting take place. However, surfing the Internet has several disadvantages too. Some notorious people misuse the Internet and hack into other peoples' private accounts. Widespread injection of Spam is also annoying.

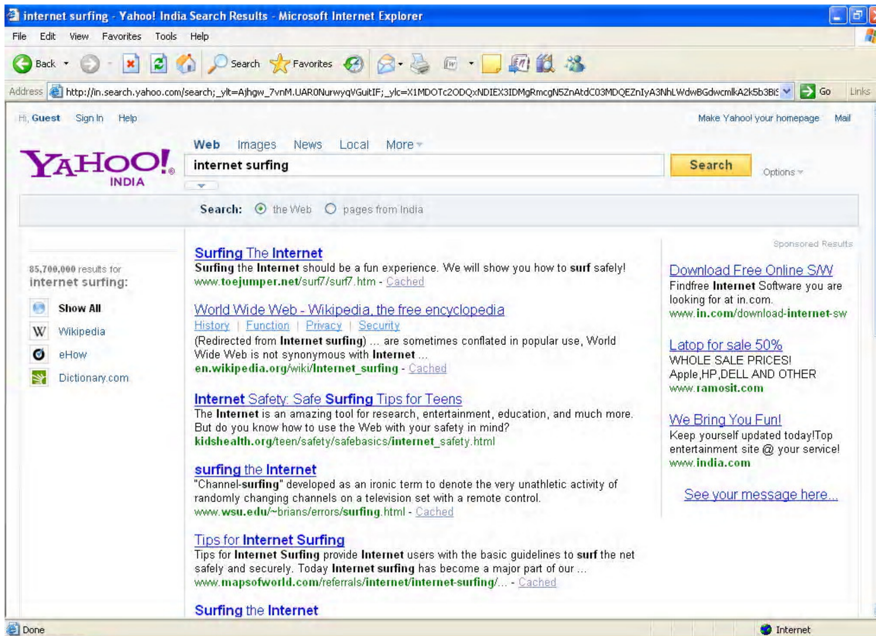
The browsing process (searching the information) is often called surfing the Internet. There is unlimited information on the Internet and it can be almost impossible to find what you are looking for. There are several ways to navigate the Web and surf the Internet. You can use a browser to visit Web sites using corresponding Web address or search engines that help you find what you are looking for. Search engines help you to find the relevant information which you require. With a search engine, you can sort a vast amount of online information. Search engines provide Web pages by matching words which you enter into the search bar. To surf the Internet with Yahoo.com, following steps are required:

- Log on to the Web and type 'http://www.yahoo.com/' in your browser's URL entry field. After pressing ENTER key, you will get the following screen:



NOTES

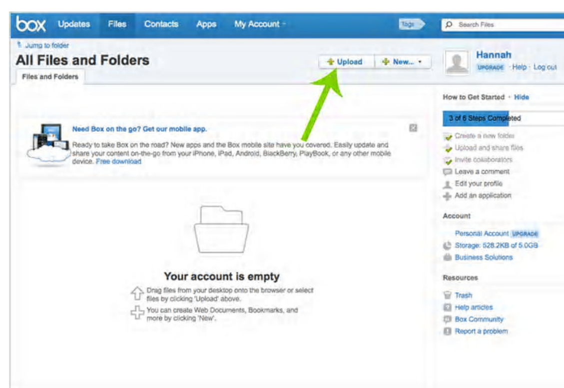
- If you want to get the knowledge about the topic ‘surfing the net’, you will get a list of searched keywords as shown in screen below.



- Click the index (searched result) provided by Yahoo and read the online information.

5.6.1 Uploading and Downloading

In computer networks, downloading means to receive data to a local system from a remote system or to initiate data transfer. Examples of a remote system from which a download might be performed are Web server, FTP server, e-mail server, etc. Downloading or uploading is relative to the system involved, for example a Personal Computer PC downloads a file from a server while the server is uploading that file to PC. A download can mean either any file that is offered for downloading or that has been downloaded or the process of receiving such a file. It has become more common to mistake and confuse the meaning of downloading and installing or simply combine them incorrectly together. Screen below displays a Web page where Upload button is provided to upload and share the content.



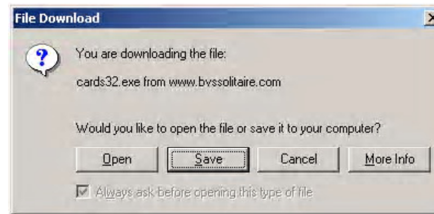
The inverse operation, uploading, refers to the sending of data from a local system to a remote system, such as a server or another client with the intent that the remote system should store a copy of the data being transferred or the initiation of such a

process. The words first came into popular usage among computer users with the increased popularity of Bulletin Board Systems or BBSs facilitated by the widespread distribution and implementation of dial-up access in the 1970s.

NOTES

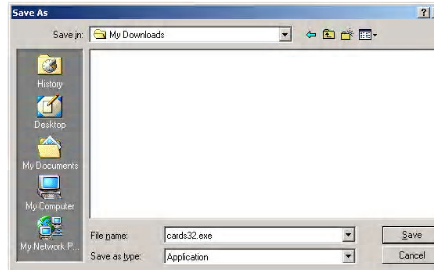
How to Download a Selected File?

Click on one of the download links on the download page. You will be presented with a dialog box which asks if you want to open the file or save it.

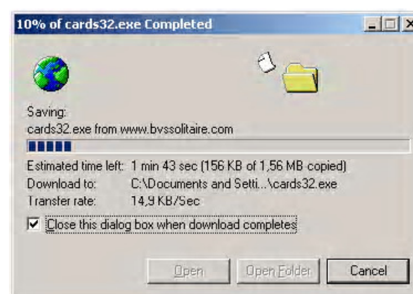


Select 'Save' button. You will next get a dialog box giving you a choice as to where the file is to be saved. You can select to have it stored anywhere on your computer. It is convenient to create temporary folder to store files you want to download. To create this folder, follow the instructions below.

- Open Windows Explorer by clicking on **Start→Programs→Windows Explorer**.
- Click on the 'C:' or other desired drive icon in the Window.
- Click on **File→New→Folder**.
- Type cards32.exe for **File name: bar** in **'My Downloads'** folder.



Click on 'Save' button. The download will now start. Your browser will tell you how much of the file has been downloaded and how much longer it will take to download the rest of the file.



Open the folder where you saved the file and install the downloaded program.

Uploading refers to sending of data from a local system to a remote system. like a server to keep a copy of the data there for various purposes. Downloading refers to retrieving of data from a remote system to a local system. Examples of remote systems include the Web server, FTP server, email server or other similar systems. Information needs to be digitized for uploading or downloading.

In the case of uploading, the files are generally copied from a smaller peripheral system to a larger central system. For example, a mobile phone file can be easily uploaded to a personal computer. Similarly, files from personal computer can be uploaded again to a server. Small files take only a few seconds in uploading, whereas larger graphic files can take hours in uploading.

NOTES

Downloading entails transfer of data from a central system to a smaller system. It is quite palpable that larger files take more time as compared to smaller files in downloading. Email is an interesting example of downloading and uploading in which emails in Inbox are downloaded from a server, whereas the replies are uploaded, so that they may be transmitted to the recipient. File Transfer Protocol (FTP) program is used to upload files to servers as well as to download files from remote locations. A number of programs are available to assist the users with uploading and downloading.

5.7 CREATING A CUSTOM INTEGRATED APPLICATION WITH MULTIPLE PROTOCOLS

A custom Web application can be created to deal with multiple Internet protocols. In this section, you will learn about the basic features of building required custom application that will help you in enhancing your own multi-protocol based applications. The application includes WebPOP which is a Web-based mail reader. The term Webmail or Web-based e-mail defines a Webmail client, i.e., an e-mail client implemented as a Web application accessed via a Web browser.

A Multi-Protocol Application

A multi-protocol application must be carefully planned before writing the program code. The following points must be checked before writing any multi-protocol application:

- Consider the functions that the application will provide and the protocols it needs to use.
- Search for similar features on the Internet that may help in fulfilling the requirements.
- Get the specifications usually an RFC or Request For Comments document of the protocols to be used.
- Specify the sockets (communication) interface of the required programming language.
- Create a draft of the overall design, such as subroutines and modules of the application.
- Code the application in the specific/required language.
- Test the application by taking special care with the subroutines that implement the different protocols.

A Multi-Protocol Application on the Web

Basically, the Web servers and Web browsers exchange information using the HyperText Transfer Protocol (HTTP), but there are several Internet services that are based on other familiar protocols, such as SMTP (mail), NNTP (news), FTP (file transfer), and

NOTES

so on. In fact, there are various techniques for exchanging information between different servers which can be placed under the same Web-based interface. The integration of several protocols in single application involves a careful design process. A general characteristic of multi-protocol Web-based applications is the use of the CGI (Common Gateway Interface) specifications. Unless an application uses a proprietary Application Programming Interface (API), it possibly uses the CGI because it is the standard way of communication between a Web server and a custom application. CGI applications can be written in almost any computer language, such as Perl and C.

By using the communication possibilities various computer languages, such as using the sockets library which are available for all major platforms, the user can write an application that converses with different servers and outputs the results in HTML format. Thus the basic feature of multi-protocol applications based on the Web includes, converse with other applications, use the information provided by them and then present the results in HTML format. The key advantage of using CGI applications is that the users have not to build a version for every known platform, because the program is executed in the server-side and also because the output is in standard HTML, plain text, etc., format. Hence, CGI is the only best alternative for creating multi-platform applications.

WebPOP



WebPOP: Common Internet protocol for retrieving mailbox contents

The use of an integrated application with multiple protocols is based on a CGI program termed as **WebPOP**, where POP stands for Post Office Protocol which is the most common Internet protocol for retrieving mailbox contents. WebPOP is a very simple mail reading program that accepts user input to present results formatted in HTML so that the user's browser can display it appropriately without any error.

WebPOP was specifically created to fulfill the requirements of the Internet users in an easy-to-use mail programs. Internet and the World Wide Web (WWW) are drawing attention of a lot of new users including the computer experts and also those which are not computer literate. These users require a very simple mail program that they can use for processing a mail messages every day. Using the standard Web browser along with WebPOP on the server-side does not require a special purpose or platform-dependent mail reader. On the client side, the only software required is a common Web browser. On the server side, the Web server must comply with the CGI specification as all the Web servers do.

WebPOP is a CGI application written in Perl (version 4.036) language which is a very useful scripting language and is specifically used for developing CGI. An entity is a program executing a particular function and is independent from other software components on the system. There are five entities that support in the functioning of the application, such as the CGI application (WebPOP), the user's Web browser, the Web server, the POP3 server and the SMTP server.

Technically, the word Web-based e-mail service specifies an e-mail service offered through a Website, such as Gmail, Yahoo! Mail, Hotmail and AOL Mail. Basically every Webmail provider offers e-mail access using a Webmail client and many of them also offer e-mail access through a desktop e-mail client using standard e-mail protocols.

Of the various applications of TCP/IP the most important one is the internetworking equivalent of the real-world postal delivery system, commonly called

electronic mail or *e-mail*. The history of e-mail goes back to the very earliest days of TCP/IP's development. Today millions of people every day send both simple and complex messages around the world through e-mail. TCP/IP e-mail is not any one application. It is implemented as a complete system comprising several protocols, software elements and components. All these elements perform one or the other part of the complete communication process of e-mail. These include a standard message format, a specific syntax for recipient addressing, and protocols to both deliver mail and allow access to mailboxes from intermittently connected TCP/IP clients.

NOTES

5.7.1 Mail Communication Process Steps

The modern TCP/IP e-mail communication process consists of the following five basic steps.

1. **Mail Composition**

E-mail journey begins with the creation of a message, that is, electronic mail message. There are two parts of a message: the *header* and the *body*. the header contains data that describes the message and controls how it is delivered and processed, the body of the message is the actual information that is to be communicated. The message must be created as per the standard message format for the e-mail system so that it can be processed. It must also specify the e-mail addresses of the intended recipients for the message.

By way of analogy to real mail, the body of the message is like a letter, and the header is like the envelope into which the letter is placed.

2. **Mail Submission**

There are various other internetworking applications besides e-mail. But, electronic mail is different from many other internetworking applications in that the sender and receiver of a message do not necessarily need to be connected to the network simultaneously, nor even continuously, to use it. The system is so designed that after composing the message, the user decides when to submit the message to the electronic mail system so it can be delivered. The mail is transferred by using the Simple Mail Transfer Protocol (SMTP).

This is analogous to dropping off an envelope at the post office, or to a postal worker picking up an envelope from a mailbox and carrying it to the local post office to insert into the mail delivery stream.

3. **Mail Delivery**

Once the user has submitted the electronic mail message, it is accepted by the sender's local SMTP system for delivery through the mail system to the destination user. Today, this is accomplished by performing a Domain Name System (DNS) lookup of the intended recipient's host system and establishing an SMTP connection with that system. SMTP also supports the ability to specify a sequence of SMTP servers through which a message must be passed to reach the desired destination. One way or the other, eventually the message arrives at the recipient's local SMTP system.

This is like the transportation of the envelope through the postal system's internal 'internetwork' of trucks, airplanes and other equipment to the intended recipient's local post office.

NOTES

4. **Mail Receipt and Processing**

Now, the local SMTP server accepts the e-mail message for further processing. It places the mail into the intended recipient's mail box, where it waits for the user to retrieve it.

In our physical analogy, this is the step where the recipient's local post office sorts mail coming in from the postal delivery system and puts the mail into individual post office boxes or bins for delivery.

5. **Mail Access and Retrieval**

The intended recipient periodically checks with its local SMTP server to see if there is any mail for him/her. If so, the recipient can retrieve the mail, open it and read its content. This is done by a special mail access protocol or method and not by SMTP. The access protocol and client e-mail software may allow the user to scan the headers of received mail (such as the subject and sender's identity) to decide which mail messages to download. This saves quite a lot of time as user need not actually open up every mail.

This is the step where mail is physically picked up at the post office or delivered to the home.

5.7.2 Electronic Mail Message Communication Model, Devices and Protocol Roles

One of the critical requirements of an electronic mail system is that the sender and receiver of a message need not be online at the time when mail is sent. TCP/IP therefore uses a communication model with several devices that allow the sender and recipient to be *decoupled*. The sender's client device spools mail and moves it to the sender's local SMTP server when it is ready for transmission. The e-mail is then transmitted to the receiver's SMTP server using SMTP where it remains for an indefinite period of time. When the recipient is ready to read it, he or she retrieves it using one or more of a set of mail access protocols and methods, the two most popular of which are POP and IMAP (see Figure 5.47).

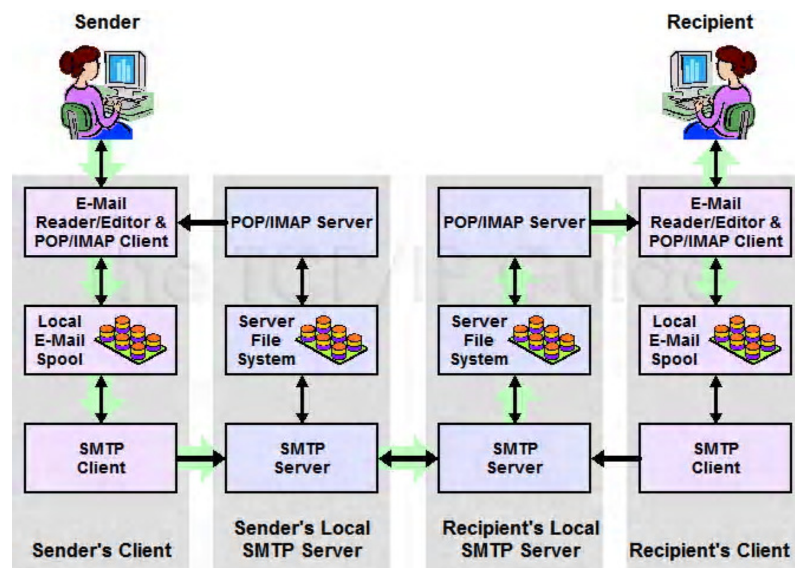


Fig. 5.47 Electronic Mail Communication Model

NOTES

- **Sender's Client Host:** The sender of the mail composes an electronic mail message, generally using a mail client program on his or her local machine. The mail, once composed, is not immediately sent out over the Internet; it is held in a buffer area called a *spool*. This allows the user to be "detached" for the entire time when a number of outgoing messages are created. When the user is done, all of the messages can be sent at once.
- **Sender's Local SMTP Server:** When the user's mail is ready to be sent, he or she connects to the internetwork. The messages are then communicated to the user's designated local SMTP server, normally run by the user's Internet Service Provider (ISP). The mail is sent from the client machine to the local SMTP server using SMTP. (It is possible in some cases for the sender to be working directly on a device with a local SMTP server, in which case sending is simplified.)
- **Recipient's Local SMTP Server:** The sender's SMTP server sends the e-mail using SMTP to the recipient's local SMTP server over the Internetwork. There, the e-mail is placed into the recipient's incoming mailbox (*inbox*). This is comparable to the outgoing spool that existed on the sender's client machine. It allows the recipient to accumulate mail from many sources over a period of time, and retrieve them when it is convenient.
- **Recipient's Client Host:** In certain cases the recipient may access his or her mailboxes directly on the local SMTP server. More often, however, a mail access and retrieval protocol, such as POP3 or IMAP, is used to read the mail from the SMTP server and display it on the recipient's local machine. There, it is displayed using an e-mail client program, similar to the one the sender used to compose the message in the first place.

Some form of addressing is required for all network communications. Since electronic mail is *user-oriented*, e-mail addresses are based on users as well. In modern TCP/IP e-mail, standard addresses consist of a *user name*, which specifies who the recipient is, and a *domain name*, which specifies the DNS domain where the user is located. A special DNS *Mail Exchange* (MX) record is set up for each domain that accepts e-mail, so a sending SMTP server can determine what SMTP server it should use to send mail to a particular recipient.

5.7.3 User Agent

The User Agent (UA) makes the sending and receiving any message easier and provides service to the user. Some examples of command-driven user agents are mail, pine, and elm. Some examples of GUI-based user agents are Eudora, Outlook and Netscape (see Figure 5.48).

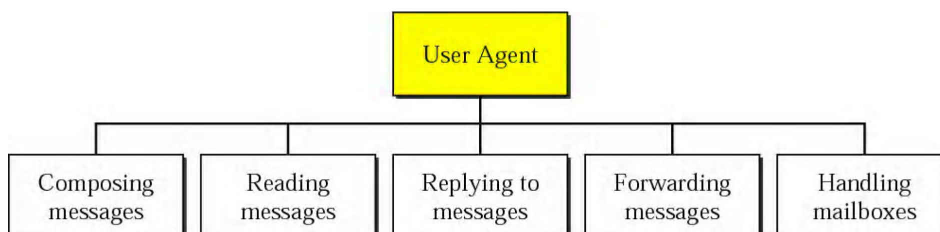


Fig. 5.48 *Functions of User Agent*

NOTES

5.7.4 Sending Mails, Receiving Mails and Addresses

1. **Sending Mails**

To send the mail, the user, through UA, creates mail that looks very similar to the postal mail (see Figure 5.49).

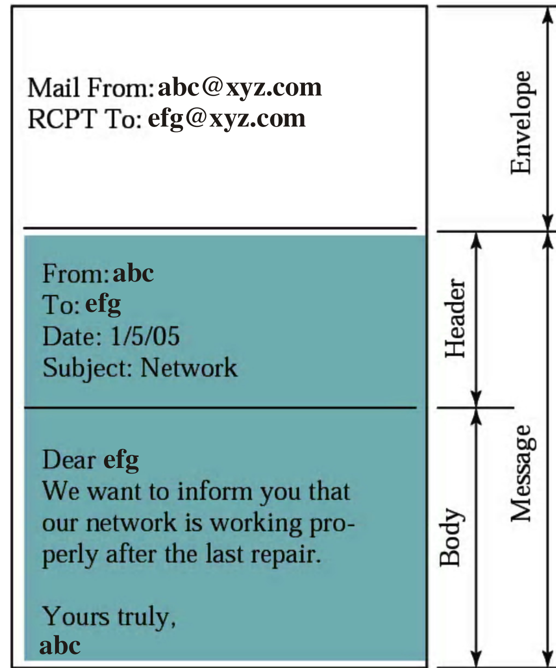


Fig. 5.49 *Format of an E-Mail*

Envelop: contains the sender and receiver addresses.

Message: contains the header and body. The header of the message defines the sender, receiver, the subject of the information and some other information. The body of the message contains the actual information to be read by the recipient.

2. **Receiving Mail**

The user agent is triggered by the user. If a user has mail, the UA informs the user with a notice. If the user is ready to read the mail, a list is displayed with each line containing a summary of the information about a particular message in the mailbox. The summary usually includes the sender's mail address, the subject and the time when the mail was sent or received. The user can select any of the messages and get the contents displayed on the screen.

3. **Addresses**

Fugyre 5.50 illustrates the address format of e-mail.

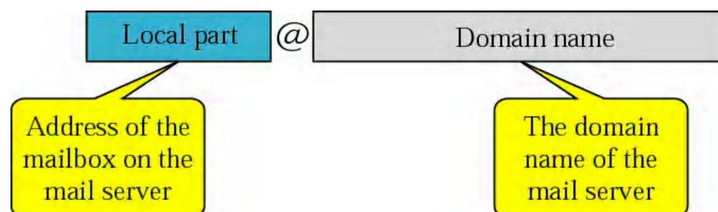


Fig. 5.50 *E-Mail Addresses*

Local Part: defines the name of the special file, called the user mailbox, where all the mails received for a user are stored for retrieval by message access agent.

Domain Name: constitutes the second part of the address. An organization usually selects one or more hosts to receive and send e-mail: the hosts are sometimes called mail servers or exchangers. The domain assigned to each mail exchanger either comes from DNS database or is a logical name.

5.7.5 Multipurpose Internet Mail Extension

The use of the RFC 822 message format ensures that all devices are able to read each other's e-mail messages, but there is one limitation: it only supports plain, ASCII text. Multipurpose Internet Mail Extension (MIME) specifies several methods that allow e-mail messages to contain multimedia content, binary files, and text files using non-ASCII character sets, all while still adhering to the RFC 822 message format (see Figure 5.51). MIME also further expands e-mail's flexibility in the information that can be carried in e-mail messages, by allowing multiple files or pieces of content to be sent in a single message. This is made possible by encoding non-ASCII data in ASCII form, and by adding special headers that describe this data and how it should be interpreted.

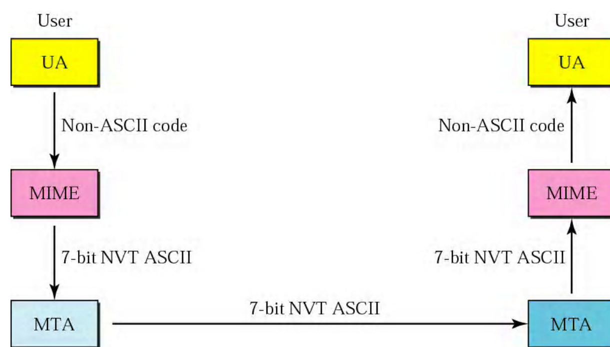


Fig. 5.51 *Multipurpose Internet Mail Extensions (MIME)*

MIME defines five headers that can be added to the original e-mail header section to define the transformation parameters (see Figure 5.52).

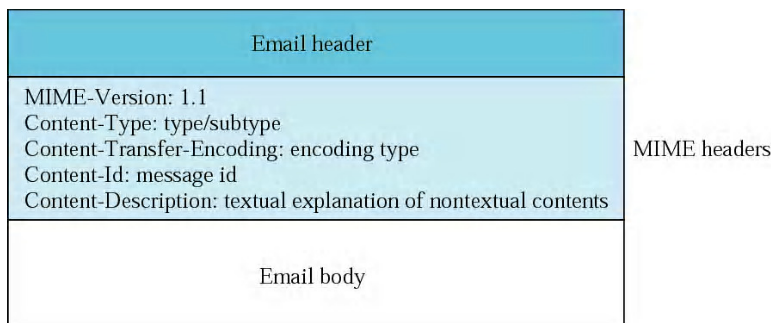


Fig. 5.52 *MIME Headers*

1. **MIME-Version**

Each MIME message is required to have a *MIME-Version* header, which serves two purposes. First, it identifies the e-mail message as being MIME-encoded. Second, even though only one version of MIME has been defined so far, having a version

NOTES

number header provides "future-proofing" in case a new version is created later that may have some incompatibilities with the present one. Right now, all MIME messages use version 1.0. This is the only MIME header that applies only to an entire message, that is, it is not used to label individual MIME body parts, and is easy to remember as it is the only header whose name does not begin with 'Content-'.

2. **Content-Type**

Content-Type describes the nature of the data that is encoded in the MIME entity. This header specifies a content *type* and a content *subtype*, which are separated by a slash character.

Content-type: `<type/subtype;parameters>`

It may optionally also contain certain *parameters* that convey additional information about the type and subtype. In a message body, this header is what tells the recipient of the e-mail message what sort of media it contains, and whether the body uses a simple or complex structure. In a body part, it describes the media type the body part contains.

For example, a message containing an HTML document might have a *Content-Type* header of 'text/html', where a message containing a JPEG graphical file might be specified as 'image/jpeg'. For a composite MIME type, the *Content-Type* header of the whole message will contain something like 'multipart/mixed' or 'multipart/alternative', and each body part will contain individual *Content-Type* headers such as 'text/html' or 'image/jpeg'.

Table 5.5 *Data Types and Subtypes in MIME*

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Chapter 22)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to Mixed, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

3. **Content-Transfer-Encoding**

For a message using simple structure, content-transfer-encoding specifies the specific method that was used to encode the data in the message body; for a composite message, it identifies the encoding method for each MIME body part. For data that is already in

ASCII form, no special encoding is needed, but other types of data must be converted to ASCII for transmission. This header tells the recipient how to decode the data back into its normal representation.

Content-transfer-encoding: <type>

NOTES

Table 5.6 *Content-Transfer-Encoding*

Type	*Description*
7bit	NVT ASCII characters and short lines
8bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data are encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters are encoded as an equal sign followed by an ASCII code

This header is optional; the default value if it is not present is '7 bit' encoding, which again is the encoding of regular ASCII.

4. **Content-ID**

Content-ID allows to assign a special identification code to the MIME content. This header is analogous to the RFC 822 *Message-ID* header field, but is specific to the MIME content itself. It is optional, and is most often used for body parts in multipart MIME messages.

5. **Content-Description**

Content-description is an optional header that allows an arbitrary additional text description to be associated with the MIME entity. In a multipart message, each body part might be given a description header to make clear to the recipient what the parts represent.

MIME supports two basic overall formats: *simple structure*, in which a single type of *discrete media* is encoded in a message; and *complex structure*, which encodes a *composite media* type that can carry multiple types of information.



Content-ID: Allows to assign a special identification code to the MIME content

5.7.6 Simple Mail Transfer Protocol

The most important component of the TCP/IP electronic mail system is the *Simple Mail Transfer Protocol* (SMTP). Derived from Mail Transfer Protocol (MTP), SMTP is the mechanism used for the delivery of mail between TCP/IP systems and users. The only part of the e-mail system for which SMTP is not used is the final retrieval step by an e-mail recipient.

In the early days of SMTP, mail was delivered using the relatively inefficient process of relaying from server to server across the internetwork. Today, when an SMTP server has mail to deliver to a user, it determines the server that handles the user's mail using the Domain Name System (DNS) and sends the mail to that server directly.

NOTES

SMTP servers both send and receive e-mail; the device sending mail acts as a client for that transaction; the one receiving it acts as a server (see Figure 5.53). To avoid confusion, it is easier to refer to the device sending e-mail as the *SMTP sender* and the one receiving as the *SMTP receiver*; these were the terms used when SMTP was originally created.

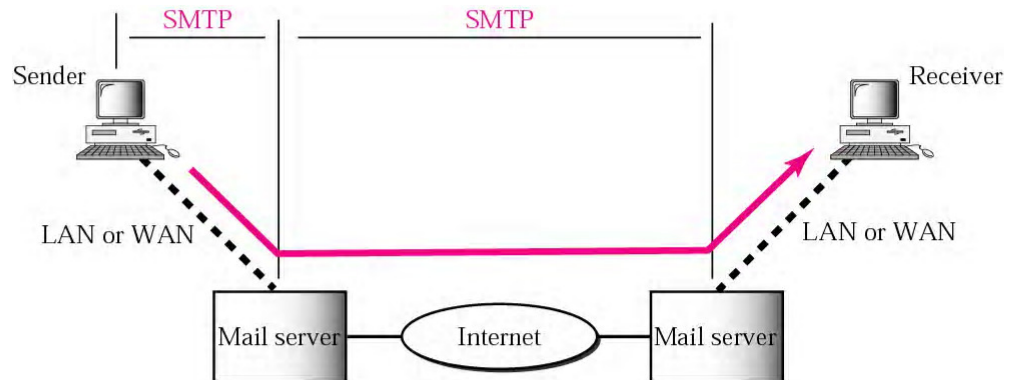


Fig. 5.53 *Simple Mail Transfer Protocol (SMTP)*

SMTP Connection and Session Establishment and Termination

The delivery of electronic mail using the Simple Mail Transfer Protocol involves the regular exchange of e-mail messages between SMTP servers. SMTP servers are responsible for sending e-mail that users submit for delivery. They also receive e-mail either intended for local recipients, or in some cases for forwarding or relaying to other servers.

An SMTP session consists of three basic phases (see Figure 5.54):

1. First, the session is **established** through the creation of a TCP connection and the exchange of identity information between the SMTP sender and receiver using the *HELO* command.
2. Once the session is established, **mail transactions** can be performed.
3. Finally, when the SMTP sender is done with the session, it **terminates** it using the *QUIT* command.

If SMTP extensions are supported, the SMTP sender uses the *EHLO* (extended hello) command instead of *HELO*, and the SMTP receiver replies with a list of extensions it will allow the SMTP sender to use.

SMTP Mail Transaction Process

The delivery of e-mail message begins with the establishment of an SMTP session between the devices sending and receiving the message. The SMTP sender initiates a TCP connection to the SMTP receiver, and then sends a *HELO* or *EHLO* command, to which the receiver responds. Assuming there are no problems, the session is then established and ready for actual e-mail message transactions.

NOTES

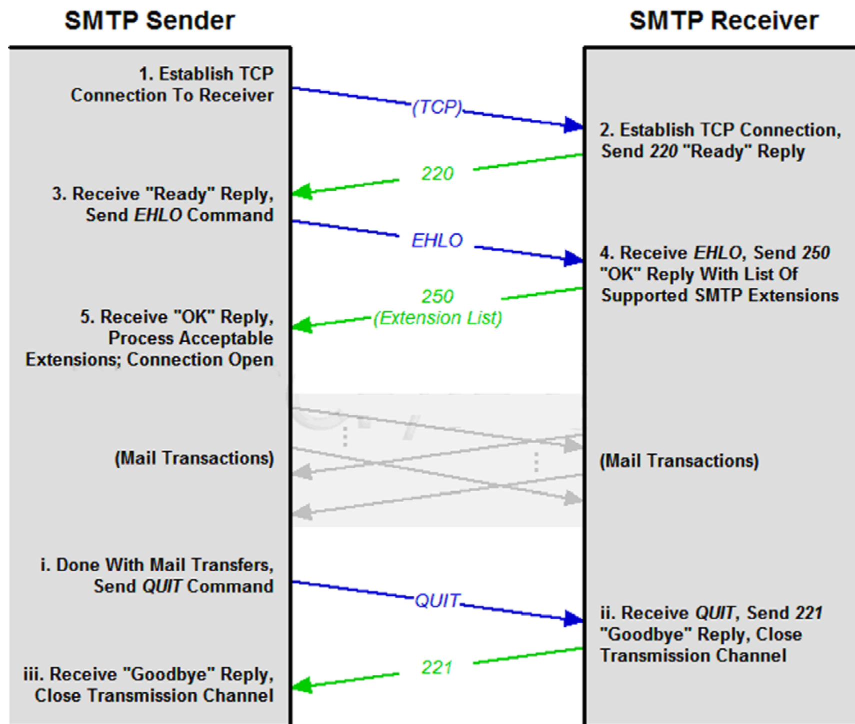


Fig. 5.54 SMTP Session

SMTP Mail Transaction Overview

The SMTP mail transaction process consists of three steps:

1. **Transaction Initiation and Sender Identification:** The SMTP sender tells the SMTP receiver that it wants to start sending a message, and gives the receiver the e-mail address of the message's originator.
2. **Recipient Identification:** The sender tells the receiver the e-mail address(es) of the intended recipients of the message.
3. **Mail Transfer:** The sender transfers the e-mail message to the receiver. This is a complete e-mail message meeting the RFC 822 specification (which may be in MIME format as well) (see Figure 5.55).

SMTP Security Issues

The base protocol does not include any security mechanism as Internet security was not an issue in the times when SMTP was designed. But with the change in current scenario, e-mail is so often abused today, most modern SMTP servers incorporate one or more security features to avoid any problem.

SMTP Commands

The SMTP sender performs operations using a set of *SMTP commands*. Each command is identified using a four-letter code. Since SMTP only supports a limited number of functions, it has a small command set (see Figure 5.56).

NOTES

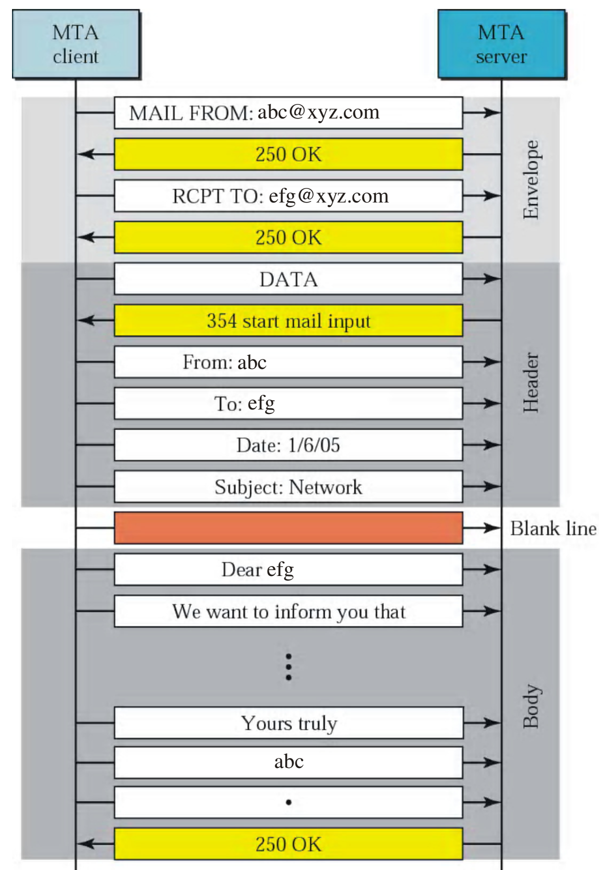


Fig. 5.55 *SMTP Mail Transaction*

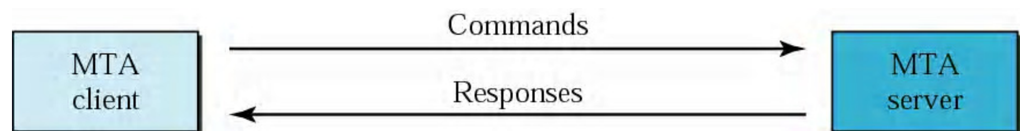


Fig. 5.56 *SMTP Commands and Responses*

- Command format: The following is the command format.

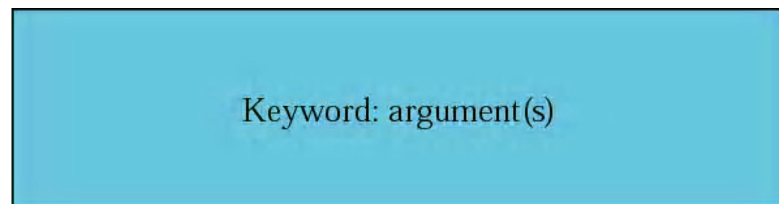


Fig. 5.57 *Command Format*

- Commands: The various commands are shown in Table 5.7.

Table 5.7 Commands

Keyword	*Argument(s)*
HELO	Sender's host name
MAIL FROM	Sender of the message
RCPT TO	Intended recipient of the message
DATA	Body of the mail
QUIT	
RSET	
VRFY	Name of recipient to be verified
NOOP	
TURN	
EXPN	Mailing list to be expanded
HELP	Command name
SEND FROM	Intended recipient of the message
SMOL FROM	Intended recipient of the message
SMAL FROM	Intended recipient of the message

NOTES

SMTP Responses

Each time the SMTP sender issues a command, it receives a *reply* from the SMTP receiver (see Table 5.8). These replies are similar to FTP replies, and uses both a three-digit reply code and a descriptive text line. A special *enhanced status codes* SMTP extension is also defined; when enabled, this causes the SMTP receiver to return more detailed result information after processing a command.

Table 5.8 SMTP Responses

Code	*Description*
	Positive Completion Reply
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
	Positive Intermediate Reply
354	Start mail input
	Transient Negative Completion Reply
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage

Check Your Progress

9. What steps should be taken to compile the ActiveX control?
10. What does built-in ActiveX controls VB offer?
11. State the format of an e-mail.
12. What does Content-Type describe?
13. What is Web- based mail?

NOTES

	Permanent Negative Completion Reply
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

5.7.7 Mail Access Protocols

For flexibility, TCP/IP uses a variety of mailbox access and retrieval protocols and methods to allow users to read e-mail. Three different models describe how these different methods work:

- The *online* model, in which e-mail is accessed and read on the server.
- The *offline* model, in which mail is transferred to the client device and used there.
- The *disconnected* model, where mail is retrieved and read offline but remains on the server with changes synchronized for consistency.

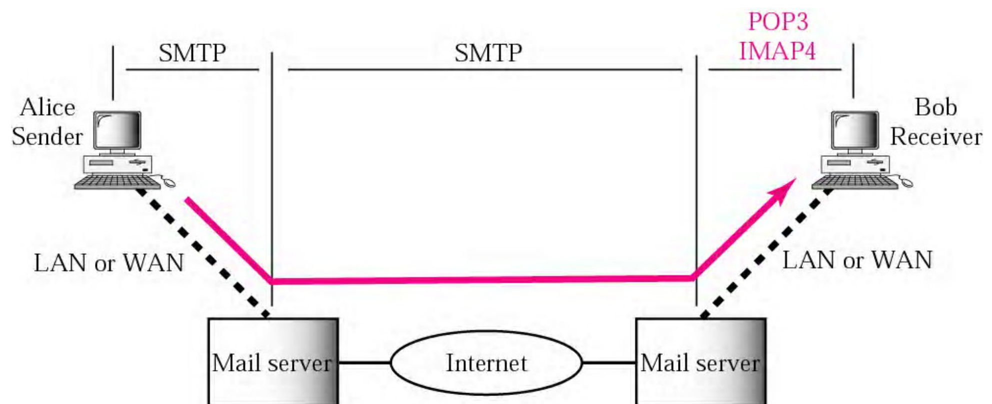


***Fig. 5.58** POP and IMAP*

TCP/IP Post Office Protocol (POP/POP3)



Post Office Protocol (POP): It is currently the most popular TCP/IP e-mail access and retrieval protocol

The **Post Office Protocol (POP)** is currently the most popular TCP/IP e-mail access and retrieval protocol. It implements the offline access model, allowing users to retrieve mail from their SMTP server and use it on their local client computers. It is specifically designed to be a very simple protocol and has only small number of commands. The current revision of POP is version 3, and the protocol is usually abbreviated *POP3* for that reason (see Figure 5.58).

- POP3 General Operation, Client/Server Communication and Session States

NOTES

POP3 is a regular TCP/IP client/server protocol. To provide access to mailboxes, POP3 server software must be installed and continuously running on the server where the mailboxes are located. POP3 uses the Transmission Control Protocol (TCP) for communication, to ensure the reliable transfer of commands, responses and message data. POP3 servers 'listen' on well-known port number 110 for incoming connection requests from POP3 clients. After a TCP connection is established, the POP3 session is activated. The client sends commands to the server, which replies with responses and/or e-mail message contents (see Figure 5.59).

POP3 is a client/server protocol that is described using a simple linear sequence of states.

1. **Authorization State:** The server provides a greeting to the client to indicate that it is ready for commands. The client then provides authentication information to allow access to the user's mailbox. By default, POP3 uses only a simple user name/password authentication method.
2. **Transaction State:** The client is allowed to perform various operations on the mailbox. These include listing and retrieving messages, and marking retrieved messages for deletion. The client normally begins by first retrieving statistics about the mailbox from the server, and obtaining a list of the messages in the mailbox. The client then retrieves each message one at a time, marks each retrieved message for deletion on the server.
3. **Update State:** When the client is done with all of its tasks and issues the *QUIT* command, the session enters this state automatically, where the server actually deletes the messages marked for deletion in the *Transaction* state. This concludes the session and the TCP connection between the two is terminated.

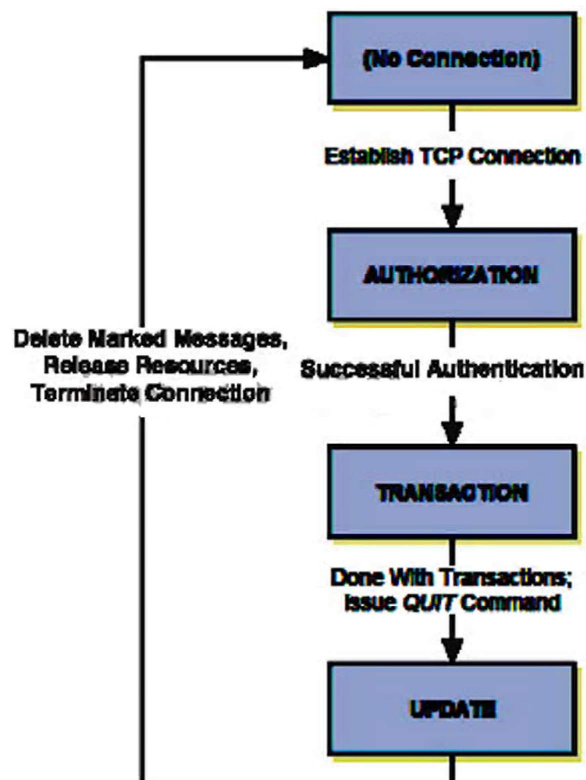


Fig. 5.59 *POP General Operation*

TCP/IP Internet Mail Access Protocol (IMAP/IMAP4)

The Post Office Protocol is popular because of its simplicity and long history, but POP has few features and normally only supports the rather limited *offline* mail access method. To provide more flexibility for users in how they access, retrieve and work with e-mail messages, the *Internet Message Access Protocol* (*IMAP*) was developed. IMAP is primarily used in the online and disconnected access models; it allows users to access mail from many different devices, manage multiple mailboxes, select only certain messages for downloading, and much more. Due to its many capabilities, it is growing in popularity.

IMAP General Operation, Client-Server Communication and Session States

IMAP is a client-server application, and an IMAP session begins with the client making a TCP connection to the server (see Figure 5.58).

The session between an IMAP4 client and server is described in the IMAP standards.

The following are the IMAP states, in the usual sequence in which they occur for a session:

1. ***Not Authenticated* State:** The session normally begins in this state after a TCP connection is established; unless the special IMAP *preauthentication* feature has been used (we will get to this feature shortly). The client at this point cannot really do much aside from providing authentication information so it can move to the next state.
2. ***Authenticated* State:** The client has completed authentication, either through an authentication process in the prior state or through preauthentication. The client is now allowed to perform operations on whole mailboxes. The client must select a mailbox before individual message operations are permitted.
3. ***Selected* State:** After a mailbox has been chosen, the client is allowed to access and manipulate individual messages within the mailbox. When the client is done with the current mailbox it can close it and return to the *Authenticated* state to select a new one to work with, or can log out to end the session.
4. ***Logout* State:** The client may issue a *Logout* command from any of the other states to request that the IMAP session be ended. The session may also enter this state if the session inactivity timer expires. The server sends a response and the connection is terminated.

5.8 SUMMARY

- Servers can be implemented as ActiveX DLL or ActiveX EXE components. The difference lies in how the server is executed.
- An ActiveX component can be created in Visual Basic by starting a new project.
- An ActiveX project is compiled in the same way as the standard `EXE` project.
- An ActiveX EXE component is an out-of process server which can be developed and run independently. It can be included in the client application after it is compiled and registered.

NOTES

Check Your Progress

14. Fill in the blanks with appropriate words.
    (a) _________ is a quite latest technology developed for supporting the broadcasting of both audio and video over the Internet or intranet
    (b) _________ video, algorithm for audio compression and the protocol that is used for the transmission of files over the Internet forms the basis of this technology
    (c) VDOLive is based on the _________ architecture where the client is VDOLive player and the server is a VDOLive server.
    (d) _________ is a cross-platform plugin architecture used by many Web browsers.
15. State whether the following statements are true or false.
    (a) Plug-ins is similar in functionality to Java.
    (b) Netscape Navigator's plug-in architecture is not based on dynamically loaded code modules.
    (c) Navigator 3.*x* enables you to stream data using Navigator plug-in. This has to be done by creating a new stream with `NPN_NewStream`, writing to it with `NPN_Write`, and destroying it with `NPN_DestroyStream`.
    (d) A multi-protocol application must be carefully planned before writing the program code.

NOTES

- Components provide reusable codes in the form of objects. An application that uses a component's code by creating objects and calling their properties and methods is referred to as a client.
- Properties for a class can be created by adding public variables and property procedures to the class module. We can also create methods for a class by adding public sub and public function procedures to the class module.
- Creating an ActiveX control is a relatively simple task in Visual Basic. For the most part, the procedure is the same as creating a standard Visual Basic executable.
- The `UserControl` Designer window displays the default project name (Project1) and control name (`UserControl1`) in its title bar. This will change once we assign meaningful names to the project and control.
- To test an ActiveX control, you need a separate Visual Basic project with a form on which you can place an instance of the control. Rather than starting a separate copy of Visual Basic, you can use one of Visual Basic's easiest features.
- When you place an ActiveX control on a Visual Basic form during program design, an instance of the control is created in memory.
- The property page interacts with the control it is attached to by using events. Whenever a property page is opened, it receives a `SelectionChanged` event.
- Once you have compiled the ActiveX control into an OCX file and removed the ActiveX project from the project group, Visual Basic will automatically switch to using the compiled version in the test project.
- A `Custom` control is an extension of the VB toolbox. A `custom` control file is a special form of a Dynamic Link Library (DLL) file that contains one or more types of controls.
- `ImageList` controls are invisible controls that serve one purpose, i.e., to hold images that are used by other controls.
- To use the images in the image list, you usually associate the image list with a Windows common control (which has an `ImageList` property).
- Status bars appear at the bottom of windows and usually hold several panels in which you can display text.
- The status bar is there to give feedback to the user on program operation as well as other items like the time of the day or key states (such as the Caps Lock or the Ins key).
- Progress bars give the user some visual feedback on what is happening during a time consuming operation.
- VDOLive is a quite latest technology developed for supporting the broadcasting of both audio and video over the Internet or intranet.

NOTES

- VDOLive is based on the client-server architecture where the client is VDOLive player and the server is a VDOLive server.
- VDOLive server runs in the background and listens to the request made to the default TCP port.
- Netscape Navigator provided a basic browser program for the Web. When Netscape first developed their Web browser, they decided they could never provide everyone who used it with every possible viewer for all files and they chose to open up their platform for outside developers.
- Netscape Plug-in Application Programming Interface (NPAPI) is a cross-platform plugin architecture used by many Web browsers.
- Plug-ins are similar in functionality to Java. In contrast to Java, Navigator plug-ins are binary code modules written and compiled for each host operating system.
- A plug-in can support more than one MIME type. This feature can be really handy for plug-ins that support more than one file format.
- Basically, the Web servers and Web browsers exchange information using the HyperText Transfer Protocol (HTTP), but there are several Internet services that are based on other familiar protocols, such as SMTP (mail), NNTP (news), FTP (file transfer), and so on.
- The use of an integrated application with multiple protocols is based on a CGI program termed as WebPOP, where POP stands for Post Office Protocol which is the most common Internet protocol for retrieving mailbox contents.
- WebPOP is a CGI application written in Perl (version 4.036) language which is a very useful scripting language and is specifically used for developing CGI.
- Technically, the word Web-based e-mail service specifies an e-mail service offered through a Website, such as Gmail, Yahoo! Mail, Hotmail and AOL Mail. Basically every Webmail provider offers e-mail access using a Webmail client and many of them also offer e-mail access through a desktop e-mail client using standard e-mail protocols.

5.9 ANSWERS TO 'CHECK YOUR PROGRESS'

1. An ActiveX DLL is an in-process server. The DLL is loaded in the same address space as the client executable that calls the server and it runs on the same thread as the client. At any given moment however, the client application or the DLL keeps running. The merit of DLL is that it is faster, as in effect, it becomes a part of the application that uses it.
2. An ActiveX EXE, otherwise called out-of-process server, runs as a separate process. When a client application creates an object provided by an EXE server for the first time, the server starts running as a separate process. If another client application creates the same object, the running EXE server provides this object. In other words, a single EXE server can service multiple clients. Out-of-process

servers seem to be more efficient in terms of resource allocation, but exchanging information between servers is a slow process.

3. The advantages of DLLs are the following:
   - The code can be easily shared among applications.
   - They offer excellent performance due to the in-process nature of the component.
   - Fixing a bug in a DLL Implement object only requires distributing an updated DLL.
   - All applications using the DLL are immediately fixed.
   - They can be used by any OLE automation client, including all VBA-based applications (such as, Microsoft Office) and other Windows development languages.
4. The following are the advantages of ActiveX controls:
   - Good performance. ActiveX controls always run in process. However, there is additional overhead involved in using an ActiveX control that does not occur with an ActiveX DLL server.
   - Controls are compatible with many containers, including Microsoft Office applications and Internet browsers.
   - Controls offer seamless integration into the VB environment.
   - Property pages allow design time user interface as well as run-time interface within Visual Basic.
   - Controls have the ability to persist design time properties in most containers.
5. The disadvantages of ActiveX control are the following:
   - Controls are considerably faster than ActiveX EXE servers, but somewhat slower than ActiveX DLL servers due to the ActiveX overhead.
   - There is some complexity involved in creating good quality controls.
   - Controls increase the complexity of deploying an application.
   - Registration, version checking and component verification are required for safe distribution.
6. An ActiveX control cannot run by itself; it needs a container. There are two choices:
   - Using Microsoft Internet Explorer or another ActiveX-capable browser as the test container. After all, one important use for ActiveX controls is on Web pages; so in some situations, testing control in a browser is perfectly appropriate. If a separate test project has not been created and the ActiveX control project is the only one loaded into the Visual Basic environment, then 'running' the project will start Internet Explorer and display an instance of the ActiveX control. The control must be compiled first. You can try out the control's capabilities as well as modify the associated HTML file if you want to test the control in combination with some scripting language elements.

NOTES

- A separate Visual Basic project can be created to test the control. This is the preferred testing method because it provides greater flexibility.

NOTES

7. When an ActiveX control is placed on a Visual Basic form during program design, an instance of the control is created in memory. When the program is run, that instance is destroyed and a new run-time instance is created; it is this instance that will be in operation as the program executes. When the program is terminated to return to the design mode, the run-time instance of the control is destroyed, and yet another instance is created and displayed on the form in the Visual Basic designer. Thus, rather than having a single instance of the control remaining, three are actually created and destroyed.
8. Property pages are useful when several properties interact in a complex fashion. You can design a property page so that related properties are grouped together, making it easier for the user to set them properly. Property pages are useful for controls that you plan to distribute internationally, because captions on property pages can easily be changed to suit different language requirements. Finally, property pages permit controls to be used with development tools that do not have Properties windows.
9. To compile the ActiveX control, follow the steps given below:
    - Ensure that all the parts of the project group are in the design mode.
    - In the Project window, click on AXCtrlDemo to make it an active project.
    - Open the File menu and select Make `AXCtrlDemo.OCX`. Visual Basic displays the Make Project dialog box. If you want the `OCX` file in a different folder, select it here. You can accept the default name for the `OCX` file, which is the same as the project name (`AXCtrlDemo`). You can also assign a different name, such as `FancyCmdButton`, if you wish.
    - Click on OK. Visual Basic will compile the project. No message is displayed upon completion, but if you look in the specified folder, you will find the `OCX` file.
    - On the File menu, select Remove Project to remove the ActiveX control project from the project group. Visual Basic will display a warning message, because the control is referenced from another part of the project group, but that is okay.
10. Visual Basic 6.0 offers us many built-in ActiveX controls such as `ImageList` control, `TreeView` control, `ListView` control, `Tab Strip` control, `Status Bar` control, etc.
11. The e-mail has the following format:

    Envelop: The envelope usually contains the sender and receiver addresses.

    Message: The message contains the header and body. The header of the message defines the sender, receiver, the subject of the information and some other

information. The body of the message contains the actual information to be read by the recipient.

12. Content-Type describes the nature of the data that is encoded in the MIME entity. This header specifies a content type and a content subtype, which are separated by a slash character.
13. In the last few years a new method has been developed to allow e-mail access using the World Wide Web (WWW). This technique is growing in popularity rapidly, because it provides many of the benefits of direct server access, such as the ability to receive e-mail anywhere around the world, while being much simpler and easier than the older methods of direct access such as making a Telnet connection to a server. WWW-based e-mail can in some cases be used in combination with other methods or protocols, such as POP3, giving users great flexibility in how they read their mail.
14. (a) VDOLive, (b) Scaleable, (c) Client-server, (d) Netscape Plug-in Application Programming Interface (NPAPI).
15. (a) True, (b) False, (c) True, (d) True.

NOTES

5.10 QUESTIONS AND EXERCISES

Short-Answer Questions

1. State the similarities and differences between DLL and EXE.
2. What is an in-process server? How does it differ from an out-of-process server?
3. List the steps involved in developing an ActiveX EXE.
4. What is instantiating? Is it used in ActiveX EXE or DLL?
5. What are the uses of ActiveX components? Mention the areas in which it is used.
6. What is ActiveX control? What are the advantages and disadvantages of using an ActiveX control?
7. What are the steps to be followed for testing an ActiveX control?
8. Discuss the different types of ActiveX controls.
9. What are `custom` controls? Are ActiveX controls and `custom` controls the same?
10. Define `TreeView`, `ListView` and `ImageList` controls.
11. What is the significant of the `PropertyBag` object?
12. Identify and list the standard controls in Visual Basic.

Long-Answer Questions

1. Develop an ActiveX DLL which calculates the train fare for the following trains:
   (i) Chenab Express
   (ii) Himalayan Queen

NOTES

(iii) Shatabdi Express
(iv) Bombay Mail

2. Explain the difference between Standard EXE and ActiveX EXE. What are the types of ActiveX components used in VB?
3. What are the steps that are to be followed for testing an ActiveX control? Explain each step with an example.
4. In what ways can an ActiveX control can be included in a project?
5. Create a user-drawn control that displays a 'star' button. Use method line to draw the button.
6. Write an 'enhanced' `TextBox` control that only accepts numeric input.
7. Discuss how an ActiveX control differs from a standard control.
8. Explain the various ActiveX controls with the help of a program to create a project.
9. Discuss the significance of VDOLive.
10. Explain the significant features of MIME, Web mail and IMAP

UNIT 6 WEB GRAPHICS

NOTES

Structure

6.0 INTRODUCTION

In this unit, you will learn about Web graphics, i.e., the various graphical applications used in Web page and Website design to give it attractive look. The computer graphics refers to the process of creating, storing and manipulating the pictures, images and drawings with the help of application software using a digital computer. With developments in computing technology, computer graphics has become an effective tool for the presentation of information in various fields including business, engineering, art, entertainment, multimedia applications, education and training. The computer generated pictures set twips, pixels and color settings for better image resolution and effective graphic presentation. You will also learn to create line, circle, box and ellipse drawings in Visual Basic for a Web page. The line control is used to draw lines on a form at design time. An ellipse refers to a closed continuous line whose points are positioned such that two points which are exactly opposite to each other have exactly the same distance from a central point.

In this unit, you will also learn about the basics of Web page and Web design. You will be acquainted with basic features of Web page, factors in creating and launching a Website, creation of home page, navigation theory in Web designing, layout of Websites, various types of Websites and architecture of Website. The Web or World Wide Web (WWW) is fundamentally an assortment of an unlimited number of pages (Web pages)/documents of information that are displayed on the Internet. It is considered as information sharing tool. A Website is a set of Web pages related to each other containing content, such as text, images, video, audio, etc. Finally, you will learn about animation. Animation in computing field refers to a process used for setting the illusion of movement where each drawing differs slightly from the earlier. The text is explained with the help of illustrations.

NOTES

6.1 UNIT OBJECTIVES

After going through this unit, you will be able to:

- Understand the significance of graphical applications in a Web page
- Draw the line, circle, box and ellipse using Visual Basic graphical properties
- Explain the significance of Web page
- Classify the document types in WWW
- Explain the significance of home page
- Create a home page for the Website
- Classify the various types of Websites
- Add graphics to a Web page
- Apply animations using Visual Basic
- Explain the usage of graphical `CommandButton`

6.2 A GRAPHIC VIEW OF THE WEB

World Wide Web (WWW) represents the network of Internetworking resources and collection of Internet sites. WWW is known as Web started in computer world in year 1989 at CERN (European Council for Nuclear Research). Web includes Web pages that are basically linked documents. These pages can be accessed across the network. When one page is linked to other, it is considered as hypertext. The word 'hypertext' was invented by Vannevar Bush in the year 1945. Getting a Web page is possible through Web browser. Many Web page editors are used to publish the Web pages, for example, Dreamweaver, ColdFusion, FrontPage, GoLive, etc. These editors are popularly used to design the Web page. Browser is the software that enables Web page to be viewed by clients. WWW or Web is a global information medium which users can read and write via computers connected to the Internet.

A Web page is an electronic document uniquely written in HTML (HyperText Markup Language). Web pages can include text, graphics, video, animation, sound and interactive elements, such as data entry forms. Web pages also include hyperlinks to other Web pages. A Website contains one or more Web pages that share to a common theme, such as a person, business, organization or a subject. The first page of the Website is called the home page, which is an index of the content available on the Website. The World Wide Web, also known as Web or WWW or W3, has established itself as the most popular feature of the Internet. It is an incredible source of information. Once you start searching anything ranging from documents to pictures to software, it almost appears limitless. It provides you with documents, sound files, view images, animation, and video, speak and hear voice, and view programs that run on practically any software in the world. Therefore, it facilitates the rich and diverse communication by enabling you to access and interact with text, graphics, animation, photos, audio and video.

Technically, the Web graphics is the methodology how to create dynamic graphics for the Website. Creating Web graphics is very easy. You can create desired

Web graphics or images in Visual Basic or Adobe Photoshop and can upload it to a Web server on the Internet. The terms image and Web graphic are often used interchangeably. Once a Web graphic or image is created and uploaded to a Web server on the Internet, it can be linked using HTML or CSS. The following are some examples of Web graphics:

NOTES

- Logos are most commonly used in Web graphics. Most Websites have a logo on every Web page. Logos are generally created using illustration software, such as Adobe Illustrator and then exported to GIF (Graphics Interchange Format) for uploading.
- Photos are also popular and are frequently uploaded as Web graphics which are captured using digital cameras and mobile camera phones. Photos are either scanned into the computer or imported directly from a digital camera. Once the photo is saved on the computer, make required adjustments in format or color, re-size it and then save it in JPEG (Joint Photographic Experts Group) format for uploading.
- Icons help a Website look more professional and also help users in fast identifying the purpose of a link. Basically, icons are created using illustration software, such as Adobe Illustrator and then exported to GIF format for uploading.
- Decorative images are also sometimes used to create the image bits that HTML and CSS can not create, for example, diagonal lines and rounded corners. Decorative images are also used for repeating backgrounds on Web pages. Depending on the image type, a decorative graphic can be created using illustration software, such as Adobe Illustrator or image editing software, such as Adobe Photoshop, Visual Basic or GIMP (GNU Image Manipulation Program) and then it is exported to GIF format for uploading.

Computer graphics refers to the creation, storage and manipulation of pictures and drawings using a digital computer. With developments in computing technology, interactive computer graphics has become an effective tool for presentation of information in diverse fields, such as science, engineering, medicine, business, industry, government, art, entertainment, advertising, education and training. There is virtually no field in which graphical displays cannot be used to some advantage and that is the basic reason why application of computer graphics is so widespread.



Computer graphics: It refers to the creation, storage and manipulation of pictures and drawings using a digital computer

It is a fact that one picture is worth a thousand words. So quite naturally interfaces empowered with graphics enhances the communication between the computer and its users. Representation of a huge set of numbers in the form of a graph or a picture helps in better understanding and interpretation of the characteristics or pattern of the data contained in the set of numbers. Graphical display helps to improve understanding of complex systems and visualization of Two-Dimensional (2-D) and Three-Dimensional (3-D) objects.

A major application of computer graphics is designing, particularly engineering and architectural systems. Almost all consumer products are now computer designed. Computer Aided Design (CAD) techniques are now extensively used in the design of building, automobile, aircraft, watercraft, spacecraft, defense mechanism, computer component, industrial machinery and device, textile and increasing number of other products ranging from a pen to a refrigerator. In computer aided design environment, traditional tools of design are replaced by parameterized modeling routines with

NOTES

interactive graphic capabilities so active and dynamic that the designer can carry out unlimited number of experiments involving complex computation in search for better design. Powerful digital graphic tools for molecular modeling have added a new dimension in the research of emerging fields like biotechnology and bioinformatics.

The applications of computer graphics, such as image processing, animation, morphing, simulation, e-learning, material designing and graphic designing are rapidly gaining demand and usage in education, training, advertisement and entertainment. Computer graphics has highly influenced the film industry with its multimedia applications. Controlled animation, simulation and morphing have increasingly been applied in the study of time varying physical phenomena, object movement and operating sequences of machinery in scientific and industrial research. Computer aided image processing and picture analysis are now indispensable tools for remote sensing, aerial survey, space research, pattern recognition, CT scans and research in medical science.

Computer graphics and animations can be created using Visual Basic. In Visual Basic 6.0, the default unit of measurement is known as twips. One twip represents 1/1440 inch or 17.639 µm. Twips represent screen independent units which ensure that the proportion of screen elements is arranged as the same on all the display systems and devices. But, a pixel is a screen dependent unit which refers to a 'picture element'. A pixel is, in fact, a dot which represents the smallest graphical measurement on a screen. Graphics are the visual presentations of real life objects. Some of the examples include photographs, drawings, line art, graphs, diagrams, typography, numbers, symbols, geometric designs, maps, engineering drawings and other images. Graphics combine text, illustration and color along with features. The graphics which are created by a computer is known as computer graphics. These graphics can be created using the representation and manipulation of image data by a computer with the help of specialized software and hardware. In these days, the computer graphics is used in almost all the fields of education and news media. Computer images and graphics are displayed on television and in newspapers, for example in weather reports, business trends, etc. A graph presents the complex statistics in a form which is easy to understand and interpret.

6.2.1 Web Graphics

Web graphics in any Websites is as significant as the content of the site. An exceptional designed graphics can present enhanced and creative ideas to the Website visitors. The Websites that do not have Web graphics can hardly appeal to visitors. Successes of Web graphics depend upon the efficient and well-organized positioning on the Websites. There is no limit in the Web specifications to the graphical formats that can be used on the Web. The following are some specific formats that can be used to store graphics.

- Portable Network Graphics (PNG).
- Joint Photographic Experts Group (JPEG).
- Graphics Interchange Format (GIF).
- Scalable Vector Graphics (SVG).

The software, such as Photoshop, Flash, Dreamweaver and Fireworks can be used for designing and creating unique graphics. Usually, graphics are used for explaining things and ideas that are not expressed easily by words, for example shopping and e-commerce Websites. The following are the basics tips for creating Web graphics:

Clarity in Web Page: In a clear Web design the things should be placed according to their importance so that the related things must look relevant with the content.

Imagery: Image plays vital role in Web designing so it is advisable to try the level best to use an appropriate image for best impact on the visitors.

Color: Every color has its own language. Hence, color combination is also very important aspects of Web designing. Without colors Web pages look dull and unattractive. Using balanced color scheme Web pages can look attractive, dynamic and inviting to users.

Readability: Font sizes used in Website plays vital role and work as great differentiators. The font size depicts something important or a new section. It is suggested to use sans-serif face for all body copy. Through using underlines, bold and italics important things can be emphasized but use them in sparing fashion.

Effective Text: HTML text is used more often in Web designing. It is more effective than the other graphic word. In the Web environment, text has enormous strengths.

Page Layout: This refers to the way in which elements are arranged on screen. So it is necessary to take great care while creating page layouts. The sequencing of contents, graphics and images play important role in conveying messages regarding products and services.

Alignment: Alignment helps to set text and graphics that should be directly, clearly and easily visible on Web pages. Left-aligned text is easier to read than right-aligned text. So, while Web designing it is suggested to follow Web alignment standard.

3D Effect on the Graphic Design: 3D effect creates space between different text and elements.

Navigation Buttons: Navigation buttons help the visitors to navigate through Website.

All these information are very important and must be followed during designing Web graphic. All graphic images on your Website should match in color, typeface and special effects.

The Role of Web Graphics

First and foremost, consistent interface and identity graphics across a collection of Web pages define the boundaries of a 'Website'. Site defining identity graphics do not need to be elaborate, but they do need to be consistent across the range of pages in a Website. The following are the various significant elements in Web graphics that serve a number of purposes as elements of content along with and complementary to text content:

- **Illustrations:** Graphics can illustrate things as per your requirements into the document.
- **Diagrams:** Quantitative graphics and process diagrams can explain concepts visually.
- **Quantitative Data:** Numeric charts can help explain financial, scientific or other data.
- **Analysis:** Graphics help in easily analysing and interpreting data.

NOTES



Page Layout: This refers to the way in which elements are arranged on screen.

NOTES

- **Integration:** Graphics can combine words, numbers and images in a comprehensive explanation.

In any document, photographs and illustrations can be concisely designed so that unnecessary explanation can be avoided and hence it saving 'thousand words' of explanation. Multifaceted information graphics often integrate quantitative charts, 3-D illustrations, etc., for explaining complex concepts and natural phenomena.

Web Graphics Basics

The following are some significant and essential elements that must be considered while creating Web graphics.

Pixels: Computer based images are comprised of thousands of tiny dots of color called pixels. These are the small dots you see if you put your face too close to your television or computer screen. Each digital image is comprised of thousands or millions of individual pixels, each with its own color. When these groups of pixels are viewed as a whole, we see the entire image. Most computer monitors have display sizes of 800 × 600 pixels, 1024 × 768 pixels or 1280 × 1024 pixels.



Resolution: Resolution indicates the number of pixels that are displayed per inch for an image (or pixels per centimeter)

Resolution: Resolution indicates the number of pixels that are displayed per inch for an image (or pixels per centimeter). Most computer monitors display at resolutions of 72 pixels per inch or 96 pixels per inch. If an image is 72 pixels wide, then it will display as 1 inch big on a monitor that is set to a resolution of 72 pixels/inch. It is important to remember that for computer graphics, a higher resolution (pixels/inch) does not mean a higher quality image. What is important is the total size in pixels of the image. Graphics for print and graphics for display on computer monitors use different resolutions. In the images for print, a higher resolution will yield a higher quality image.

Graphics and File Size: While crating Web graphic the image quality and file size must be considered. Generally, the higher the quality of the image, the greater the file size will be. Also, the larger the image, the larger the file size will be. It is important to consider the file size of images when using them on a Web page. Images with large file sizes take longer to load which can be an annoyance to the visitor.

Essay Web Graphics

Planning essays with a clear and coherent paragraph structure and flow is a challenging skill. Providing visual ways of constructing and representing an essay structure can be very helpful to many students, particularly those who are visual learners.

The essay Web is a graphic depiction (see Figure 6.1) of a structure for an essay. The Web design is used to show how each paragraph is connected to each other paragraph and how each paragraph is connected to the overall contention. The circular movement may provide an alternative concept to the usual common procedure of a linear set of steps and help students see how the essay has unity.

Russell Kealey describes a two stage process for essays where several filmic devices are compared across three different films. Firstly students complete a three set Venn diagram to help identify which filmic devices are present strongly in all three films. This thinking helps the students populate an electronic essay plan set up in Publisher with movable text boxes. In each text box, the students enter dot points about how (say) filmic device A is used in film B. Further text boxes call for introductory, linking

and concluding sentences and then the whole set can be rearranged into a visual plan for the essay.

Another advantage of these sorts of graphics is that they allow students to think about one aspect of the essay plan without having to hold all the other aspects in their short term memory.

NOTES

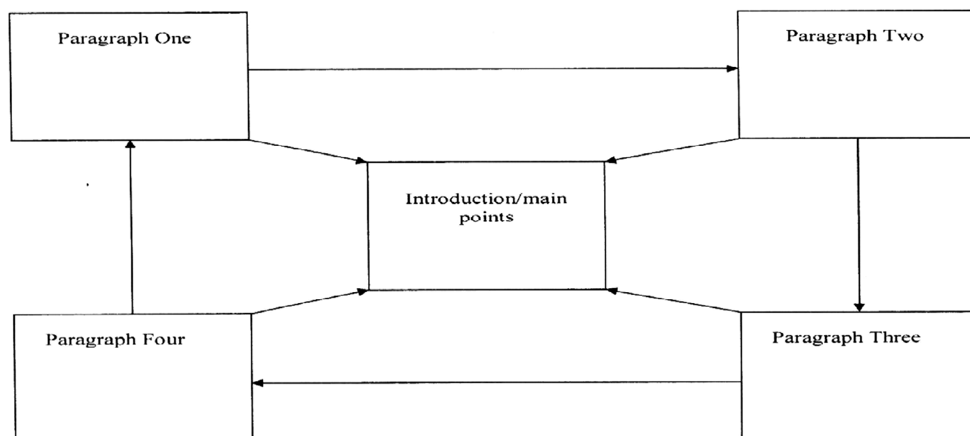


Fig. 6.1 Web Graphic Depiction

6.2.2 Images and Hyperlinks

The terms image and Web graphic is often used interchangeably on the Web. Once you have created a Web graphic and uploaded it to a Web server on the Internet, you can then link to it using HTML or CSS.

The HTML specification does not prescribe or limit which graphics format you can use. Graphics may be associated with an HTML document in the standard ways. There are three ways to include inline objects in HTML. The `IMG` element is the most common method for using graphics in HTML pages. For faster display, the width and height of the image can be given as attributes. One attribute that is required is alt, used to give an alternate textual string for people browsing with images off or who cannot see the images. The `OBJECT` element in HTML can contain other elements nested within it, unlike `IMG` which is empty. This means that several different formats can be offered, using nested `OBJECT` elements, with a final textual alternative (including markup, links, etc.) right at the center. The `APPLET` element is used for embedding applets into the HTML page. These applets can do many things but the common task is to use them to display images, particularly ones in unusual formats or which need to be presented under the control of a program for some other reason.

Background Image using CSS: Any HTML element can (when displayed visually) be given a background image using Cascading Style Sheets (CSS). The background color can also be specified and the image will composite onto the background color if it has transparent portions. CSS does not prescribe or limit which graphics format you can use.

The XML specification does not prescribe or limit which graphics format you can use. It also does not provide a standard method to describe image inclusion. W3C defines standard hyperlinking capabilities including links to graphics to XML. This is similar to hyperlinking capabilities of HTML.

NOTES

Any XML element can (when displayed visually) be given a background image using Cascading Style Sheets (CSS). The background color can also be specified and the image will composite onto the background color if it has transparent portions. This allows the same image to be re-used with different visual presentations. The width and height of the image can be adjusted. It can repeat in the horizontal(x) direction, vertical (y) direction, both or neither. The position of the image relative to the box taken up by the element can be adjusted. CSS does not prescribe or limit which graphics format you can use.

Graphics as Hyperlinks

Once the Web graphic is created you can insert graphics on your Web page using hyperlink. The following is HTML code that displays graphics that act as hyperlinks.

```
    <a href="valid Web address">
    <img src="graphic file name">
</a>
```

Some common uses for hyperlink are as follows:

- Linking to the Website's home page via a graphical heading or logo appearing at the top of each Web page.
- Linking to the next page, previous page or the top of the current page via 'button' navigational links at the bottom of each page.
- Linking to all the pages in a Website via buttons in an index or menu.
- Linking to large full-sized graphics via thumbnail or miniaturized versions.

6.3 SITE AND PAGE DESIGN

The WWW is a subset of the Internet and comprises of a huge collection of documents stored in computers across the world. The Web encompasses special sites called Websites along the Internet that support Web browsing. By clicking on the links that appear on the Webpage, one can navigate from one place to another. Hence, Webpage can be defined as a single hypertext document written in HyperText Markup Language (HTML) and described in HTML basics. A Web page normally incorporates the basic information and links to navigate in the Websites to which it belongs. Documents in the World Wide Web are classified into three types, namely static, dynamic and active documents.

Static Web Page

These are fixed content documents which perpetually provide the same information in response to all download requests from all Web users. Static documents are stored in a Web server to be accessed by the Web client. The Web client, on requesting for a Web page gets a copy of the same. The contents of such files are not subject to modification on part of the Web user as the Web user does not have right to alter them. However, the Web pages can be modified in the server per say. Thus, the static Web pages display the same information to all the Web users and provide hypertext links to perform navigation through static documents. Their biggest advantage is that they are cache friendly. This enables the Web pages to display one copy of the same

Web page to many people simultaneously. However, it becomes difficult to maintain Web pages in case of large sites as they demand consistency and updation.

Dynamic Web Page

NOTES

These Web pages provide interactive Web navigation and help modify the content like text, images, form fields, etc., on a Web page, depending on different contexts or conditions. The dynamic Web pages make use of two types of inter-activities, which are enlisted as follows:

- **Client Side Scripting:** It is used to modify interface behaviours within a specific Web page. This modification is based on the mouse or keyboard actions and is conducted at specified time intervals. The dynamic behaviour takes place within the presentation. The presentation technologies like JavaScript or ActionScript for Dynamic HTML (DHTML) and Flash for media types are used. The client side scripting also facilitates the use of remote scripting in which the DHTML page requests for additional information from the server. The content is generated on the Web client's machine in which the Web browser retrieves a page from the server and processes the code embedded in the Web page, so that the contents of the retrieved page can be displayed to the Web user. Sometimes, the Web browsers do not support the language and the commands of the scripting language, in the client side dynamic pages.
- **Server Side Scripting:** It is used to modify the requested Web page source amongst pages to either adjust the sequence or reload the Web pages delivered to the browser. Server responses are based on certain conditions like data in a posted HTML form, parameters in the URL, the type of browser being used, the passage of time or a database or server state. Server side scripting dynamic Web pages are designed with the help of server side languages like PHP, Perl, ASP, JSP, etc.



Client side scripting: It is used to modify interface behaviours within a specific Web page

Both the techniques may be used simultaneously to develop the dynamic Web pages. The advantages of dynamic Web pages are that these facilitate easy update of the Web pages and faster Web page loading. In the dynamic Web pages, the content and the design are located separately, thereby allowing frequent modifications to the Web pages including the text and image updates.

Active Documents

The programs that run at the client side are known as the active documents. Whenever a Web client requests for an active document, the Web server provides a copy of the same in the form of byte code. The document is now ready to be run at the Web client machine. As the active document is served in the binary form, compression and decompression can be applied at the server and the client side to reduce the bandwidth requirement and throughput.

6.3.1 Web Design

Home page is known as the first page of the Web page. It is replete with a myriad of hyperlinks on its page. Creation of a home page connotes creating and launching of the Website. This is a consequential task which is accomplished by arranging the Website hosting, designing and coding of the Website, monitoring the functioning of the site and by scrutinizing the Website traffic. Creating the Website takes into consideration, various factors which are to be implemented on the page. Launching

NOTES

the Website is an important operation. This requires information pertaining to name, phone, URL description as well as the domain details. It should be ensured that the Website must be kept in the right direction. A comprehensive user's guide that conveys the relevant information of the Website must be provided to the user. This can be done after successfully launching the Website. The more accurate the details are, the better the results would be. The task of launching the Website can be carried out in the local listing of Google and Yahoo. This optimizes the search engine facilities for your Website, which offers moderate list of options, searchable information and the third party data providers, such as SuperPages, YellowPages, CitySearches, etc. These search engines provide a great facility to recite the name of your Website. Such search engines also offer a free Jumpstart program in which you can enter you Website domain area, avail the 'WAY' (Who You Are) facility, get reviews and list hours, etc. There are many factors that determine the success of a Website on the Internet. The following factors should be taken into consideration while creating and launching a Website:



Message board: It is a type of forum through which visitors of the Website interact with the site to enhance its popularity

- **Message Board:** It is a type of forum through which visitors of the Website interact with the site to enhance its popularity.
- **Search Engine:** This is a valuable retention tool which helps visitors to search relevant information, provided the site is enlisted with a famous search engine.
- **Polls:** This option on the Website enables the visitors to vote as per their satisfaction. For instance, the users can assess the performance of the Web services by giving their feedback with the help of the feedback option.
- **Guestbooks:** This Website facility helps the users contact the organization for which the particular site is created. Through this facility, the Website visitors can enter their name, e-mail, comments and suggestions. Once this information reaches the organization, the respective executive of the organization contacts the visitors.
- **Data Entry Forms:** Through this option, the Website visitors can place orders and can also provide request information. Data entry forms also facilitate storing of customer service data which is later entered into a computerized database or spreadsheet by the organization.

Customer Information	
Customer No.	
First Name	
Last Name	
Sex	
Address Line 1	
Address Line 2	
City	
State	
Post Code	
Country	
	Submit to database or Add another customer record?

Fig. 6.2 Data Entry Form for Customer Record

Figure 6.2 depicts the data entry form for customer record that provides various text fields, submit buttons, links to navigate to another page, combo boxes, etc. If you click on the link 'Add Another Customer Record?' as specified in Figure 6.2, it will provide another set of customer record fields to fill the detail of customer information.

NOTES

Creation of a Home Page

Creation of a home page requires eight steps which are as follows:

1. **Select and Register a Web Page Domain Name**

   First select a suitable Website domain name to monitor the conflict issues of the same. Once a domain name is allotted to an organization or an individual, it can not be further allotted to anyone. The registration of domain name is unique and is carried out by Internet Corporation for Assigned Names and Numbers (ICANN), which is an accredited domain name registrar, such as abc.com, xyz.com, etc. The free Website hosting service is also available that can be availed without registering a domain name. The search engine does not provide its services if any Website lacks its registered domain name.

2. **Select and Configure a Website Hosting Service**

   The hosting cost of Website ranges from $100 to $250 every year. The cost varies from one Website to another, depending on the Websites' features, such as e-commerce facilities, special processing requirements, high traffic volume options, etc. At this stage, Web hosting is checked for control over content, security and usage of the site.

3. **Design, Code and Test the Website**

   A static Website comprises of a single Web page. It must have 'index.html' or 'index.htm'. A bare-bone format of the Web page which constitutes the HTML code, is as follows:

   ```
    <HTML>
      <BODY> Hello Web! </BODY>
   </HTML>
   ```

   There are various types of software tools, such as Adobe Photoshop, Microsoft FrontPage, etc., for designing a Website.

4. **Deploy the Website to the Host**

   At this stage, the file transfer program is uploaded in order to download the Website. It also updates the pages between system units and Website host computer.

5. **Security and Authentication**

   Before launching a Website on the Internet, it is essential to implement the security and foolproof authentication of the site. The following security methods are required for the same:

   - Login pages should be encrypted.
   - Data validation should be conducted in the server side.
   - Managing the Website with the help of encrypted connections.
   - Websites must be connected using secured network.
   - Login credentials must not be shared.
   - Maintaining a password and key authentication.
   - Using redundancy to protect the Website.

NOTES

After accomplishing the authentication only SSL (Secure Sockets Layer) is used with **http://URL.** The login form POST is encrypted after every login process. The JavaScript is used for validating the Web forms. The server side validation safeguards from malicious security crackers but in a limited way. The Website must be equipped with encrypted connections because non-encrypted connections make the Website susceptible to login or password sniffing or even man-in-middle-attacks.

A secured connection must be associated with Website and also secured proxy server must be used, such as **PuTTY** secure proxy or **OpenSSHproxy**. In order to maintain secure workstation, integrity auditing must be conducted. The server failover and backups should also be deployed as they diminish the possibility of server crashes. Data backup is also important so far as it safeguards from loosing the client's data.

6. **Online Payment Mode**

Before launching any Website, it is essential to set up an online payment mode which is provided to the Website. A Website must be equipped with facilities like payment through credit cards or by a third-party, such as PayPal, etc.

Other factors that should be considered while launching a Website are as follows:

- A Website is launched with the help of File Transfer Protocol (FTP). It is an economical option. The owner of Website must instruct the Web designer and the system analysts to implement FTP before the launch of the Website.
- The Web hosting firm place on the server must be provided on the site. For example, site owner provides the disk to the Web host firm so that they can set up fee for the site.
- It is incumbent upon the owner of the Website to remove the 'teething' problem before launching the Website. Incoherent or incomplete Website can discourage the visitors. For example, if an organization provides e-commerce services, it must ensure that up to date and relevant information is available on its site. The teething problem may lead to problem of set up and layout of the Web screen.



Website: It is written in HTML and is a collection of linked Web pages on a Web server which can be electronically accessed

A **Website** is written in HTML and is a collection of linked Web pages on a Web server which can be electronically accessed. Web server is a machine in which a Website is located or hosted. It may be organization owned or Internet Service Provider (ISP) owned. The Web pages may be owned by a university, a private company or an individual and are accessible to all people. Most of the Websites have their own homepages that facilitate navigation by providing links to explore the details stored therein. The pages of a good Website are organized using a common theme.

7. **Launching the Website**

Launch of the site is carried out after designing and completion of the site. It is essential to finalize the layout and style of the site before launching. It is significant to note that before the launch of the Website, its domain should be registered.

8. **Promote the Website**

The information is sent on the Web through search engines and their related directories. The promotion scheme must be published on the Website at regular

intervals. Therefore, this factor must be considered during creation of the Website. The optimal way to promote the Website is to update the visitors on the specific Website with the pertinent information. For example, in case of online air ticket booking systems, any promotional scheme, such as shifting the seat arrangement from economic to business class or changing the flight schedules etc. must be updated online to intimate the travellers and the visitors about the same.

NOTES

It provides a point of entry to a Website with help. It also contains all relevant links of a particular Website, so as to enable the user to explore the Website for information available therein.

6.3.2 Types of Website

A Website is a collection of related Web pages containing images, videos or other digital files. It is hosted on at least one Web server, accessible via a network, such as the Internet or a private local area network through an Internet address known as a Uniform Resource Locator or URL. A Web page is a document written in plain text and interspersed with formatting instructions of HyperText Markup Language or HTML, eXtensible HyperText Markup Language or XHTML. A Web page may incorporate elements from other Websites with suitable markup anchors. The user's application, often a Web browser, renders the page content according to its HTML markup instructions onto display terminal. The pages of a Website can usually be accessed from simple URL called the home page. Active documents are sometimes referred to as client side dynamic documents. Active documents can be created in following ways:

- Java Applets, programs written in Java on the server, are compiled and ready to run. The browser creates an instance of this applet and runs it.
- JavaScript is interpreted and run by the client at the same time. The script is in the source code.

Static and dynamic Websites are the types of Websites which are discussed below:

Static Websites

A **Static Website** is one that has Web pages stored on the server in the format that is sent to a client Web browser (see Figure 6.3). It is primarily coded in HTML. This type of Website includes simple information about a company and its products and services via text, photos, animations, audio and video and interactive menus and navigation. This type of Website usually displays the same information to all visitors.



Static Website: It is one that has Web pages stored on the server in the format that is sent to a client Web browser

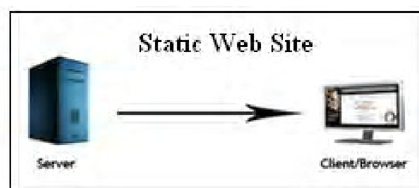


Fig. 6.3 *Layout of a Static Website*

Dynamic Websites

A **Dynamic Website** is one that changes or customizes itself frequently and automatically based on certain criteria. Dynamic Websites can have two types of dynamic activities namely, code and content. Dynamic code is invisible or behind the scenes and dynamic content is visible or fully displayed. Figure 6.4 displays the layout of a dynamic Website.



Dynamic Website: One that changes or customizes itself frequently and automatically based on certain criteria

NOTES

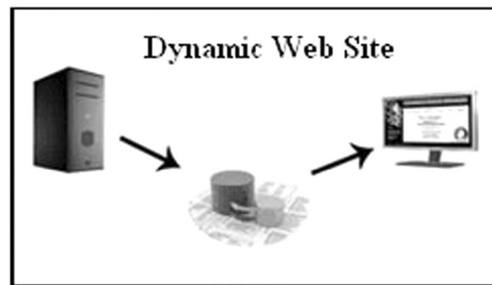
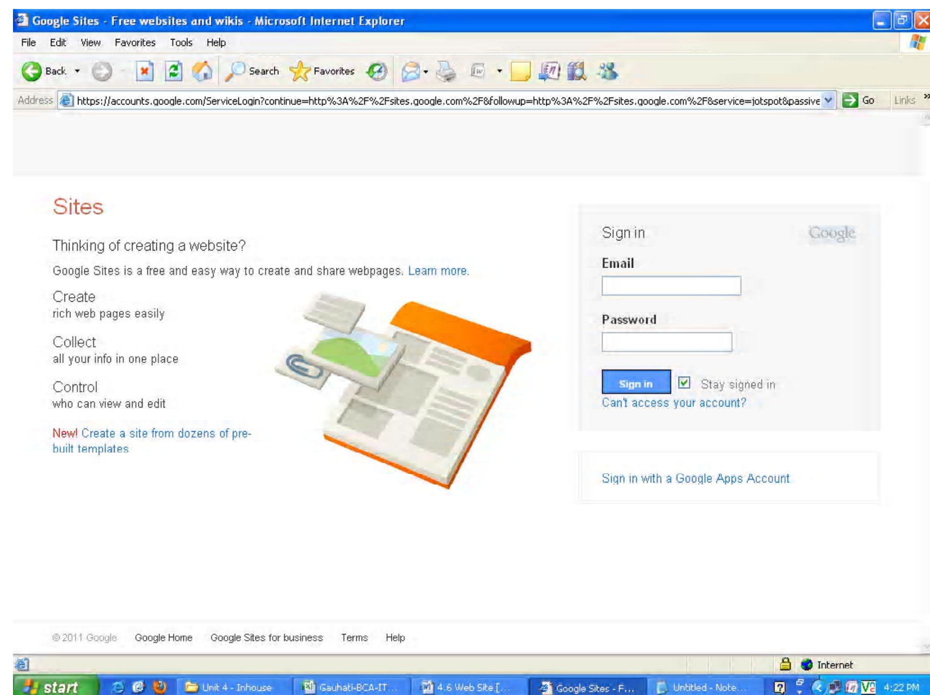


Fig. 6.4 *Layout of a Dynamic Website*

In Website, keywords are the words which are used to surf the corresponding information regarding Web page content. Keywords are important because it allows your customers to find your Web page when they search the Internet. Bounce rates can be used in determining the effectiveness or performance of an entry page of Website. An entry page with a low bounce rate means that the page effectively causes visitors to view more pages. Google Analytics is a free service offered by Google that generates detailed statistics about the visitors to a Website. Google Sites is a free and easy way to create and share Web pages.



The WWW was originally designed as a stateless entity. A client sends a request; a server responds. In Figure 6.5, Registrar is the company of domain name. Web address, such as 'www.yourdomain.com' hosting company represents the company where files, data and information is kept for a specific Website (see Figure 6.5). Web designer represents the development team who creates the Website (Web pages, scripts, page layout and graphics) whereas the Internet (connecting all the Web servers across the Internet).

NOTES

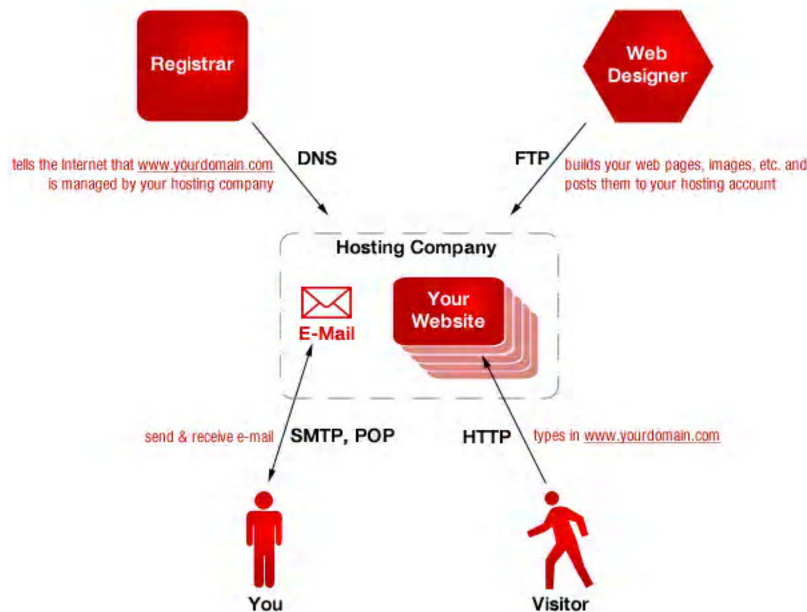


Fig. 6.5 *Architecture of Website*

Navigation Theory and Website

Navigation theory and practice for Web designing represents some prime and specific factors. These factors are known as link, URL, bookmarks and HTML tags. The site maps and Three-Dimensional or 3-D visualizations come to the hypertext navigation which provides complete information that can be availed by the users if they click on address bar by selected phrases or keywords. In fact, links are considered as the best possible way to practice the navigation theory. Basically, navigation can be defined by the HTML tags that let users to jump from one page to another page. The site is divided into content areas in which navigation theory provides to the users to access each of the content frequently. The main benefit of this technique is incorporated into the navigation into a paragraph text and hypertext links. The navigation practice is preferred to the Website because of the following reasons:

- This mechanism provides users a rough idea about the scope of the desired Website.
- This mechanism sometimes referred to fallback mechanism in high-traffic across network so that if users try to access the searched information the required information can come frequently.
- This mechanism reduces the garbage information if users collect and merge the information.
- It always provides a Home menu that describes the whole information about the created Website.
- This mechanism saves the time of users by providing the regular links.

Site tagged with navigational systems are logically hyperlinked so that users hover the mouse on the image link or click on the text that must be intuitive and clear to them.

Page Layout of Proposed Website

The Website template provides a page layout for practicing the navigation theory. Most of the page layout is required for the multimedia applications and navigation link,

NOTES

for example, a header, a left column with the navigation, a right column with the main content and footer as shown in Figure 6.6.

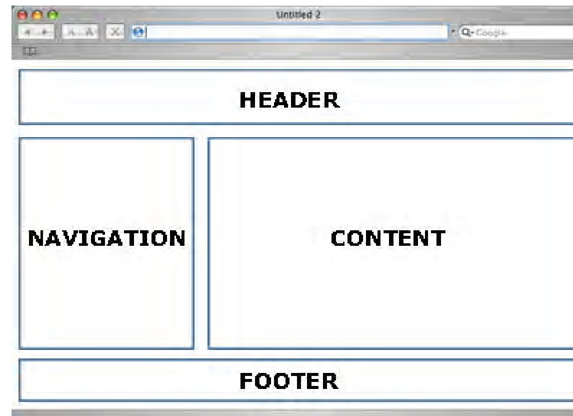


Fig. 6.6 Layout of a Website

The content and navigation are ensured suitable page footer in two ways, such as using the styles in Cascading Style Sheets or CSS and tables. CSS is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language. The most common application is to style Web pages written in HTML and XHTML but the language can also be applied to any kind of eXtensible Markup Language or XML document, including plain XML, Scalable Vector Graphics or SVG and XML User Interface Language or XUL. The CSS ensures a suitable page header and below that navigation is set in the left side column as shown above. A primary navigation must be always clear, structured and intuitive. It holds at least 4 to 15 links in which group of 4 to 5 links are enclosed at a time along with headings so that navigation bar perform fast tasks after scanning or clicking. The secondary navigation keeps the common place links, such as Contact, About Us, Site Map, Home and copyright information. Links between the pages of the Websites are considered as certain way to tie the multiple pages together. Some of the features, such as 'Link Back to Your Homepage' is always considered as essential link because the home page is considered as central hub which is used to connect almost all the navigational links, 'breadcrumb trail' shows frequently the row of links which represents the how the Website is structured whereas 'Page jumps' represents the link of sections in a single page. In 'Page jumps' format, a long page is maintained so that user can reach easily to required bookmark. A 'site map' is also a significant mechanism which links to all the pages to the Home page links.

Navigating the Web

Navigating the Web means moving from one Web page to other. Web page is designed generally in that way in which whole information might come within a page. Generally Web page is aligned on the left side. The purposes of navigating the Web are as follows:

- It presents visitors along with most user friendly path so that they can get quick information via requested page.
- It endures visitors to know the location of Website.
- It allows visitors to move quickly on logically arranged pages.

- It gives visitors suitable context of the document which users get.
- It highlights the classification of organization that has to be promoted.

Features of navigation used in navigating the Web pages in proposed Website are as follows:

- **Central Navigation:** This part contains main body of information. For example, Google: 'Search engine, Web and Protocol', etc., is treated as central navigation.
- **Global Navigation:** It contains hyperlinks which provide flexibility of accessing almost every page linked on the site, such as Contact us, Home, etc.
- **Feature Navigation:** It provides attractive content on site for example, promotion scheme feature or breaking news, etc.
- **Content Navigation:** It occurs on specific document. If one document refers to other document, a link is made.
- **Drop-Down Navigation:** It delivers drop-down menu and used as a space saver.
- **URL Navigation:** This represents an address of the requested page consisting of communication protocols. This address is followed by colon and two slashes. An example of URL is 'http://'. The full URL alerts you the URL structure.
- **Text Links:** These links are underlined by different color rather than text. If the image or graphics is hyperlinked, the moving cursor around it shows the address of site. The arrow cursor takes a shape of hand. URL is appeared in the status bar at the lower left position in the browser.

NOTES



Drop-down navigation: It delivers drop-down menu and used as a space saver

Links

Links appear on Web page such that they navigate the corresponding Web pages as a reference. If user navigates the pages to search or get the information through links that is also called in Web technology as hyperlink provides a very quick journey of Web pages. Link is same as citation in literature. It is of two types known as text link and image link. Text link provides a string based link to users. If users click on the anchored link or clickable text, they can easily get the required information which reside on the specific link. For example,



Links: It appear on Web page such that they navigate the corresponding Web pages as a reference

Customer Service (URL: http://www.My_Company_Ad.com**)**

In the above link, the customer service URL is made as text link. It will provide the information of Website which is specified as a Web address.

Types of Website

The types of Website are as follows:

- **Commercial Website:** Commercial Website especially in B2B or Business-to-Business but increasingly in B2C or Business-to-Consumer is when people want information about your business, your products or services and they expect to get that information from your Website. The purpose of this type of Website is to sell products or services. Commercial Websites are used for promoting a business or service and are among the most common type of Website on the

NOTES

internet. The Internet address often ends with '.com'. Commercial Websites operate as online businesses.

- **Personal Website:** The purpose of this type of Website is to provide information about an individual or group. This type of Website contains information or any content that the individual wishes to include.
- **Organizational Website:** The purpose of this type of Website is to support an individual's opinion or a group's point of view. The organizational Website is communicated with each other usually by chat or message boards. The Internet address often ends with '.org'.
- **Educational Website:** The purpose of this type of Website is to provide information about an educational establishment or to present information in an educational manner. The Internet address ends with '.edu'.
- **Entertainment Website:** The purpose of this type of Website is to entertain and provide amusement. The Internet address often ends with '.com'.
- **News Website:** The purpose of this type of Website is to provide information about current events dedicated to dispensing news and commentary. The Internet address often ends with '.com'.
- **Hybrid Website:** Many Websites are a mixture of different types of sites. For example, a business Website may promote the businesses products, but may also host informative documents, such as white papers or provide news for the visitors. There are also numerous sub-categories to the ones listed above. For example, there would be a hobby site where the Webmaster shares their knowledge with other likeminded individuals and through forums or chat rooms create a dialogue with them.
- **Dynamic Websites:** Dynamic Websites allow you to make your own changes to one or more sections of your Website from an Internet connection on any computer worldwide. Through a password protected administrative area, you can easily add, edit or delete content, pictures and links. Changes appear on the Website immediately. Dynamic Websites communicate with a database that pulls content into the page when a link is clicked. This is the preferred application for e-commerce, membership sites, event calendars, mailing lists and sites where frequent changes are needed.
- **E-commerce Websites:** E-commerce has exploded across the business world like few other technological developments in history. While a standard store front serves pass-by traffic and a certain geographic area with overhead costs and staffing requirements and only during business hours, an online store is open 24×7 for 365 days a year and a potential customer base of millions. It accepts credit card payments, have a catalog of your products available online, process orders automatically, increase the visibility of your brand, track shipments and market your company to the far reaches of the WWW that means to your business.



- **Real Estate Websites:** These Websites help to sell the residential property or several commercial office blocks.
- **Non-Profit Websites:** It is being offered on the Web server for all non-profit organizations our Web design services at an incredible discount of approximately 30 or 40 per cent the normal rate. It could be a charity organization, a public service group or a small business.
- **Blog:** A Website that is used to log online readings or to post online diaries may include discussion forums or chat rooms. The Internet address has a variety of endings. A **blog** is a type of Website or part of a Website supposed to be updated with new content from time to time. Blogs are usually maintained by an individual with regular entries of commentary, descriptions of events or other material, such as graphics or video.



NOTES



Blog It is a type of Website or part of a Website supposed to be updated with new content from time to time

Website Architecture

The layout and design framework of a Website and each of its pages describe collectively to the Website architecture. The relationship between each page and the whole Website as well as the relationship between each page and every other page is the primary consideration when designing a Website from an architectural standpoint. How both content and navigation are positioned on each of the pages is also an extremely important consideration both for the end-users visiting the site and for search engine optimization. It displays the building blocks and architecture of successful Website. The architecture is divided into five layers which are Impact Layer, Surface layer, Persuasion Layer, Value Creation Layer and Management Layer. In Impact layer, the key success indicators are trust and credibility, branding and image, user experience, loyalty and retention and conversion rates. Surface layer deals Home page, landing pages, navigation layout and content information architecture. The features of successful Web page, such as marketing and promotion personas, buying behaviours/motivations, conversion funnels and persuasion strategies are designed in Persuasion layer. In Value

NOTES

Creation layer, relationship building optimization strategies, pricing offer and positioning competitive intelligence are designed whereas iterative management, tracking and measuring unique value proposition, strategic purposing goals and objectives are designed in Management layer. Site identify, branding, content layout and design, eye catching presentation of Websites are designed in two layers, such as Surface layer and Persuasion layer .

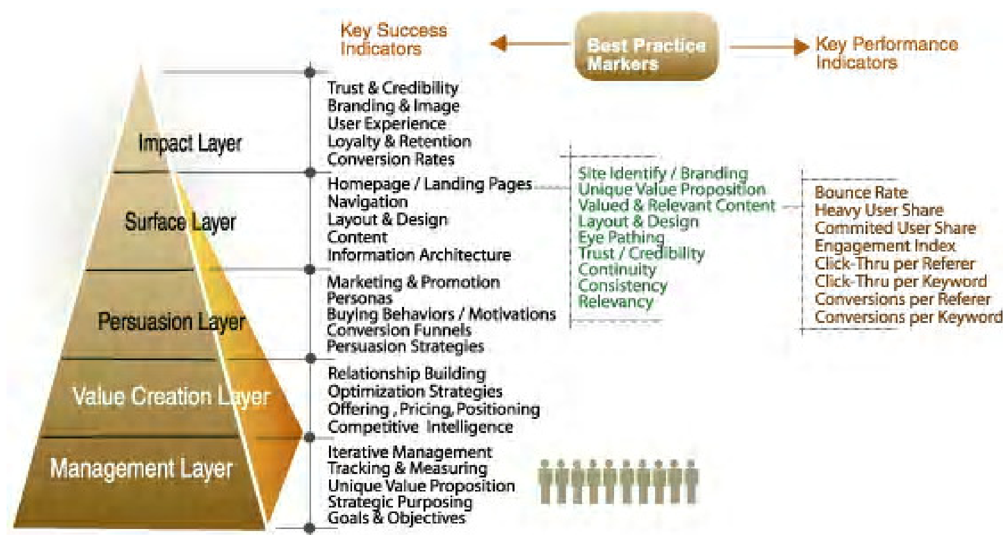


Fig. 6.7 Building Blocks and Architecture of Successful Website

The basic Web architecture is characterized by two-tiered services in which Web client gets the information and Web server provides information to the client. The Website architecture depends on key standards, such as HTML used for encoding the content of document, Web browser is used to search the information, HTTP is used for staging the transfer, Hypertext and Links is used the Motion of the Web, pages on the Web (represents the layout of Web pages searched via various search engines), URIs is used for naming the remote information objects to the global namespace, CGI for three-tier architecture. The Common Gateway Interface or CGI extends the architecture to three-tiers by adding a back-end server that provides services to the Web server on behalf of the Web client, permitting dynamic composition of Web pages.

Web Hosting

A **Web hosting service** refers to a type of Internet hosting service that allows individuals and organizations to make their own Websites accessible via the WWW. Web hosts can be understood as the companies that provide space on a server they own for use by their clients and provide the Internet connectivity. Web hosts can also supply the datacentre space and connectivity to the Internet for servers they do not own to be located in their datacentre, called collocation. The various types of Web hosting services can be summarized as follows:



Web hosting service: Refers to a type of Internet hosting service that allows individuals and organizations to make their own Websites accessible via the WWW

- **Free Web Hosting Service**: Such services are generally supported by advertisements. They often have limited services as compared to paid hosting.
- **Shared Web Hosting Service**: In case of a shared Web hosting service, one's Website is placed on the same server as many other sites, ranging from a few to

hundreds or thousands. In such a case, all domains may share a common pool of server resources, such as RAM and the CPU.

NOTES

- **Reseller Web Hosting**: In this kind of Web hosting service, clients may themselves become Web hosts. Also, resellers could function, for individual domains, under any combination of the listed types of hosting, depending on who they are affiliated with as service providers.
- **Virtual Dedicated Server:** A server is divided into many virtual servers so that each user feels as if it is his own dedicated server, which ideally is not the case, i.e., he is sharing a server with many other users. With regard to security measures, such service providers may provide root access to respective users their own virtual space only.
- **Dedicated Hosting Service:** In this type of service, the user gets his/her own Web server and gains full control over the same (root level access for UNIX/ Linux and administrator access for Windows). However, it should be noted that the user typically does not own the server. Another feature of dedicated hosting is that it is self-managed or unmanaged.
- **Managed Hosting Service:** In this kind of service, the user gets his or her own Web server but is not allowed full control over it. He is allowed to manage his data via FTP or other remote management tools. He is disallowed full control so that the provider can ensure quality of service by not allowing the user to modify the server or potentially create configuration problems. The server in this case is leased to the client.
- **Collocation Web Hosting Service:** It is similar to the dedicated Web hosting service, but in this case, the user owns the collocated server. The hosting company service provider provides the physical space which is needed by server and then takes care of the same. This is the most powerful and expensive type of the Web hosting service. The client typically would have his own administrator visit the datacenter on site to do any hardware upgrades or changes. The collocation provider may provide little to no support directly for their client's machine, providing only the electrical, Internet access and storage facilities for the server (which usually vary among service providers).
- **Clustered Hosting:** In such type of hosting, multiple servers host the same content for the better utilization of resources. The various types of clustered hostings provided by the Web host service providers are as follows:
  - File hosting service (host files not Web pages)
  - Image hosting service
  - Video hosting service
  - Blog hosting service
  - One-click hosting
  - Shopping cart software

Table 6.1 lists the Web hosting providers ranked by the best price–value ratio. Host reliability, uptime, key features, bonus features, customer support, past and current user feedbacks, user-friendliness and hosting awards have been taken into account for ranking them.

NOTES

Table 6.1 *Top Ten Web Hosting Providers*

Rank	Web hosting provider	Features
1	Inmotion: Business Web hosting	Free domain name, host 6 domains, top technical support
2	Bluehost: Unlimited Web hosting	Free domain forever, host unlimited domains, $75 marketing bonus
3	WebHostingPad: Cheap, reliable hosting	Free domain forever, host unlimited domains, $100 marketing bonus
4	JustHost: Unlimited Web hosting	Free domain forever, host unlimited domains, $50 marketing bonus
5	Hostmonster: Cheap Unix hosting	Free domain forever, host unlimited domains, $75 marketing bonus
6	Globat: Cheap Web hosting	Free domain name, host unlimited domains, $55 marketing bonus
7	Yahoo: Reliable Web hosting	Free domain name, $100 Yahoo credit, $50 Google credit
8	GoDaddy: Largest Web hosting	Free domain name, host unlimited domains, free SSL certificate
9	Lunarpages: Cheap, shared hosting	Free domain forever, host unlimited domains, free $775 software
10	Dot5Hosting: Cheap Unix hosting	Free domain forever, host unlimited domains, $75 marketing bonus

6.4 ADDING GRAPHICS TO WEB PAGES

In order to draw the primitive objects, one has to first **scan convert** the objects. This refers to the operation of finding out the location of pixels to be intensified and then setting the values of corresponding bits, in the graphics memory, to the desired intensity code. Each pixel on the display surface has a finite size depending on the screen resolution and hence a pixel cannot represent a single mathematical point. However, we consider each pixel as a unit square area identified by the coordinate of its lower left corner, the origin of the **reference coordinate** system being located at the lower left corner of the display surface. Thus each pixel is accessed by a non-negative integer coordinate pair (X, Y). The X values start at the origin and increase from left to right along a scan line and the Y values (i.e., the scan line numbers) start at bottom and increase upwards.

Line drawing is accomplished by calculating intermediate point coordinates along the line path between two given end points. Since screen pixels are referred with integer values, plotted positions may only approximate the calculated coordinates, i.e., pixels which are intensified are those which lie very close to the line path if not

Check Your Progress

1. Name the major application of computer graphics.
2. What does 'twips' represent?
3. Define the term pixel.
4. What are dots?
5. What is aspect ratio?
6. Why the `ClipControls` property is used?
7. How line drawing is accomplished?

exactly on the line path, which is in the case of perfectly horizontal, vertical or 45° lines only. Standard algorithms are available to determine which pixels provide the best approximation to the desired line. Still, screen resolution is a big factor towards improving the approximation. In a high resolution system the adjacent pixels are so closely spaced that the approximated line pixels lie very close to the actual line path and hence the plotted lines appear to be much smoother almost like straight lines drawn on paper. In a low resolution system, the same approximation technique causes lines to be displayed with a "stair step appearance" (refer Figure 6.8(a)).

NOTES

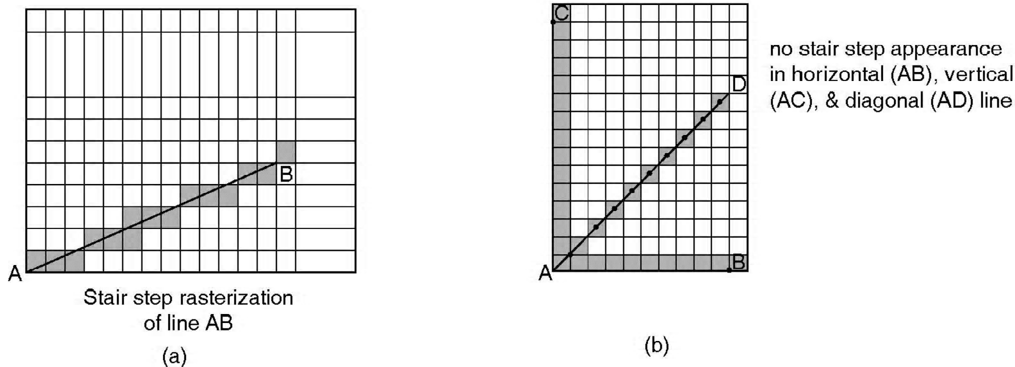


Fig. 6.8 *Line Drawing*

Following is the list of objects used in Visual Basic for graphical representation:

- **Graphics Object:** Provides methods for drawing a variety of lines and shapes. These methods provide various properties for making simple or complex shapes in solid or transparent visibility, allows filling of content with a range of colors using user defined gradient or image textures.
- **Pen Object:** To draw lines and curves.
- **Brush Object:** To fill in a closed area with some color or texture.

Table 6.2 summarizes the list of colors and their constants which can be used in Visual Basic code.

Table 6.2 *List of Colors Constants*

Color Constants	Color
vbRed	
vbGreen	
vbBlue	
vbCyan	
vbMagenta	
vbYellow	
vbBlack	
vbWhite	

NOTES

To Draw a Line

In Visual Basic, the `Pen` object is used to draw lines between specific points. This object is also used to define the specific widths. To draw a line, the code is written as follows:

```
Dim redPen As New Drawing.Pen(Color.Red, 10)
e.Graphics.DrawLine(redPen, 20, 30, 100, 100)
```

Above code draws a line between the specified coordinates with a width of 10 pixels as follows:



To Draw a Circle

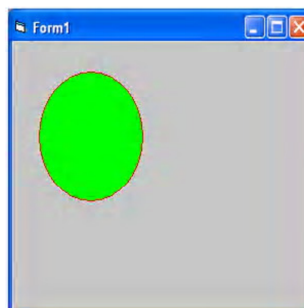
In Visual Basic, drawing circle specifies the circle's centre and its radius. Following Visual Basic code is written to draw a circle with a radius of 1000 twips in the upper left corner of the form:

```
Circle (1200, 1200), 1000
```

In Visual Basic, the `Circle` method uses the current values of the `DrawWidth`, `DrawStyle`, `FillStyle` and `FillColor` properties which collectively draw the circles with thick borders and can also fill them with a pattern. The circle's border is drawn using the current `ForeColor` value but you can override it by passing a fourth argument as defined in `Circle` method:

```
' A circle with a 3-pixel wide red border
' Filled with green solid color
DrawWidth = 3
FillStyle = vbFSSolid
FillColor = vbGreen
Circle (1200, 1200), 1000, vbRed
```

Above code draws a circle 3-pixel wide red border and filled with green solid color:



NOTES

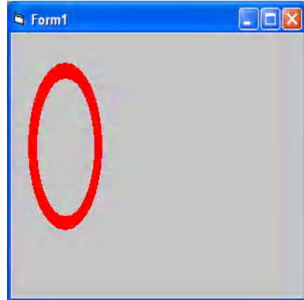
To Draw an Ellipse

To draw an ellipse, use the `Circle` statement. It takes two arguments that determine the shape of the ellipse, namely the radius and he aspect ratio. The aspect ratio represents the number and can be calculated when you divide the *Y*-radius by the *X*-radius of the ellipse.

To draw an ellipse using `Pen` object, the code is written as follows:

```
Dim redPen As New Drawing.Pen(Color.Red, 10)
e.Graphics.DrawEllipse(redPen, 10, 10, 100, 200)
```

Above code draws an ellipse using a width of 10 pixels as follows:

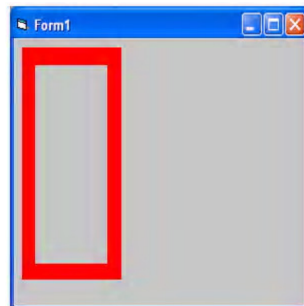


To Draw a Rectangle

To draw a rectangle, the code is written as follows:

```
Dim redPen As New Drawing.Pen(Color.Red, 10)
e.Graphics.DrawRectangle(redPen, 10, 10, 100, 200)
```

Above code draws a rectangle using a width of 10 pixels as follows:



To Draw a Box

A box is drawn in Visual Basic using the same snytax as drawing a rectangle. The `Paint` event subroutine is provided by an interface of the coding Editor. For this, you need to create a paint brush with which you paint using the following set of statements:

```
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs)
    Handles Me.Paint
    Dim redBrush As New Drawing.SolidBrush(Color.Red)
End Sub
```

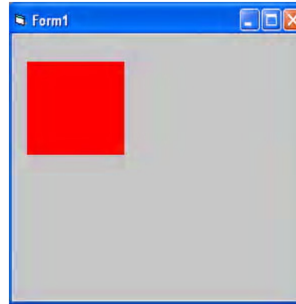
In the above code, `SolidBrush` drawing object is created using color Red and assigned it to variable `redBrush`. Then, the brush is used to draw a shape:

```
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs)
                    Handles Me.Paint
```

```
    Dim redBrush As New Drawing.SolidBrush(Color.Red)
    e.Graphics.FillRectangle(redBrush, 20, 30, 100, 100)
End Sub
```

NOTES

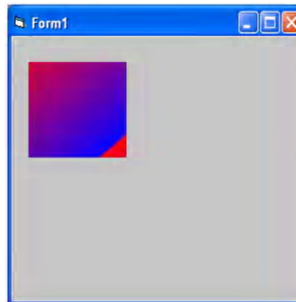
The above example uses the `Graphics` object to draw a rectangle at coordinates *X*=20, *Y*=30 of height and width equal to 100 for each. Press F5 function key to build and run the application which appears as follows:



Visual Basic supports various brush effects, such as gradient effect. The gradient effect is achieved using the `LinearGradientBrush` method which is written as follows:

```
Dim myRectangle As New Drawing.Rectangle(10, 10, 100, 100)
Dim myGradient As New Drawing2D.LinearGradientBrush
(myRectangle, Color.Red, Color.Blue, 50)
e.Graphics.FillRectangle(myGradient, 20, 30, 100, 100)
```

The above code produces the following output:



Check Your Progress

8. What is a Web page?
9. What is a static Web page?
10. Define the term Website.
11. What is a blog?
12. The basic Web architecture consists of how many tiers?
13. Write the significance of animation in graphics.
14. What does `Enable` property of `CommandButton` determine?
15. What is the simplest way of creating animation?
16. How `Image.Left` and `Image.Top` give the distance of the image in twips?
17. What does the `Font` property produce in a `CommandButton`?
18. Why is `ImageIndex` property used in `CommandButton` graphics?

6.5 DYNAMIC GRAPHICS–ANIMATION

Literally speaking, to *animate* is to bring to life, i.e., to put something into action. Animation makes graphics more realistic by imparting motion and dimension to an inanimate object. Intuitively though we think of animation synonymous with motion, technically speaking, it covers all changes that have a visual effect. Thus it may include time varying position (motion dynamics), shape, size, color, texture (update dynamics) of an object and also changes in lighting, camera position, focus, etc.

Application

With advancement in computer aided techniques, today animation is extensively used in Entertainment (games and movies), Educational and Training presentations, Advertising, Internet and Process simulation. Process simulation through animation is very useful in visualization of functioning and stages of operations of industrial products (like a gear or motor) or gradual transformations in a complex process like changing atomic structures in a chemical reaction or distortion of structures under dynamic forces.

NOTES

Elements

A computer animation sequence can be set up by specifying the storyboard, the object definitions and the image frames. The ***storyboard*** is an outline of action. It consist of rough sketches of motion sequence or it could be a list of basic events that take place. ***Object definitions*** are given for each participating object in terms of their shape and movement. The still image frames are either drawn manually or computer generated to simulate motion sequence of animating objects. The illusion of movement is created by playing 15-20 numbers of such still images with small changes made to each one per second. The eyes retain an image long enough to allow the brain to connect the frames in an uninterrupted sequence. In traditional animation, as many as 30 FPS (Frames Per Second) might be used to give a smoother appearance at high speeds.

Animation Techniques in Visual Basic

Animation is an interesting and exciting part of programming. Even though VB is not specifically designed to handle advance animations still some simple interesting animated effects can be created. The simplest way to create animation is to set the Visible Property of a group of images or pictures or texts and labels to True or False by triggering a set of events, such as clicking a button.

Example

The following is a VB program that creates the illusion of moving the plane in four directions, i.e., North, South, East and West. To do this animation, first insert five images of the same picture of plane into the VB Form. Set the Visible property of the image in the center to be `True` while the rest set to `False`. On start up, a user will only be able to observe the image in the center. Next, insert four command buttons into the Form and change the Labels to Move North, Move East, Move West and Move South, respectively. Double click on the Move North button and key in the following procedure:

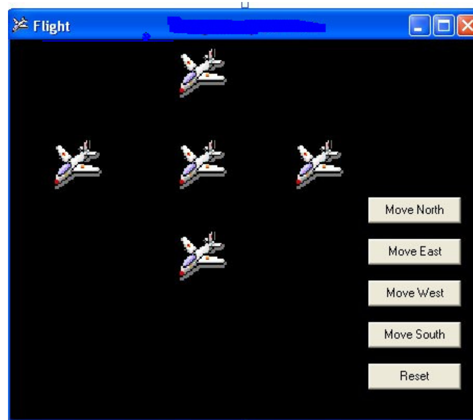
```
Sub Command1_click( )
Image1.Visible = False
Image3.Visible = True
Image2.Visible = False
Image4.Visible = False
Image5.Visible = False
End Sub
```

By clicking on the Move North button, only `Image3` is displayed. This will give an illusion that the plane has moved in North direction. Enter similar procedures by double clicking other command buttons to complete the rest of the actions. You can also insert an additional `CommandButton` and label it as Reset and then enter the following codes:

```
Image1.Visible = True
Image3.Visible = False
Image2.Visible = False
Image4.Visible = False
Image5.Visible = False
```

Clicking on the Reset button will make the image in the center visible again while other images become invisible, this will give the false impression that the plane has moved back to the original position as shown in the following screenshot.

NOTES



You can also issue the commands using a `TextBox` as illustrated below:

```
Private Sub Command1_Click()
If Text1.Text = “n” Then
Image1.Visible = False
Image3.Visible = True
Image2.Visible = False
Image4.Visible = False
Image5.Visible = False
ElseIf Text1.Text = “e” Then
Image1.Visible = False
Image4.Visible = True
Image2.Visible = False
Image3.Visible = False
Image5.Visible = False
ElseIf Text1.Text = “w” Then
Image1.Visible = False
Image3.Visible = False
Image2.Visible = False
Image4.Visible = False
Image5.Visible = True
ElseIf Text1.Text = “s” Then
Image1.Visible = False
Image3.Visible = False
Image2.Visible = True
Image4.Visible = False
Image5.Visible = False
End If
End Sub
```

Another simple way to simulate animation in VB6 is by using the `Left` and `Top` properties of an object. `Image.Left` gives the distance of the image in twips from the left border of the screen and `Image.Top` gives the distance of the image in twips from the top border of the screen, where 1 twip is equivalent to 1/1440 inch. Using a statement, such as `Image.Left-100` will move the image 100 twips to the left, `Image.Left+100` will move the image 100 twips away from the left (or 100 twips to the right), `Image.Top-100` will move the image 100 twips to the top and `Image.Top+100` will move the image 100 twips away from the top border (or 100 twips down).

6.5.1 Graphical CommandButton

NOTES

Graphial `CommandButton` refers to a button with a rectangular border that contains text, a graphic or both. When user clicks on a `CommandButton` to specify a command, it initiates an action. Table 6.3 summarizes the `CommandButton` and `Label` properties.

Table 6.3 CommandButton and Label Properties

Property	Description
Name	The Name property represents the name of the object so you can call it at runtime.
BackColor	The BackColor property specifies the Command button's background color. To set this property, you need to click on the BackColor's palette to see a list of common Windows control colors where you can change this to the style property from 0 to 1.
Cancel	The Cancel property determines whether the CommandButton gets a Click event if the user presses Esc key.
Caption	The Caption property holds the text that appears on the CommandButton.
Default	The Default property determines if the CommandButton responds to an Enter key which sends control over to the specific event.
Enable	The Enable property determines whether the CommandButton is active. You can also change the Enable property at runtime with code to prevent the user pressing the button.
Font	The Font property produces a Font dialog box in which you can set the caption's font name, style and size.
Height	The Height property positions the height of the specified object.
Left	The Left property positions the left control for the specified object.
MousePointer	If MousePointer is selected to an icon, it can change the picture of the mouse pointer over that object.
Picture	The Picture property holds the name of an icon graphic image so that it appears as a picture instead of a button for this option to work the graphical tag must be set to 1.
Style	The Style property determines if the Command button appears as a standard windows dialog box or a graphical image.
Tab Index	The Tab Index specifies the order of the command button in tab order.
Tab Stop	The Tab Stop specifies whether the object can be tabbed to. This property also used in labels which have no other function.
Tool Tip Text	The Tool Tip Text property is used to display a brief description about the action, for example if the mouse is held over the object (pictures or icons), a brief description is displayed.
Visible	The Visible property is set as true when you want the user to see the button or label select true otherwise false is selected.
Width	The Width property is used to show the width of the object.

To load a picture or image into an image box or a picture box, you can click on the `Picture` property in the Properties Window and a dialog box will appear which will prompt the user to select a certain picture file. You can also load a picture at runtime by using the `LoadPicture()` method. The syntax is as follows:

```
    Image1.Picture= LoadPicture("C:\path name\picture file name")
picture1.Picture= LoadPicture("C:\path name\picture name")
```

For example, the following statement will load the grape.gif picture into the image box.

```
    Image1.Picture= LoadPicture("C:\My Folder\VB
program\Images\grape.gif")
```

NOTES

In the following example, each time you click on the 'Change Pictures' `CommandButton`, you will be able to see three images loaded into the image boxes. This program uses the `Rnd` function to generate random integers and then uses the `LoadPicture()` method to load different pictures into the image boxes using the `If...Then` statements based on the random numbers generated. The output is shown as follows:

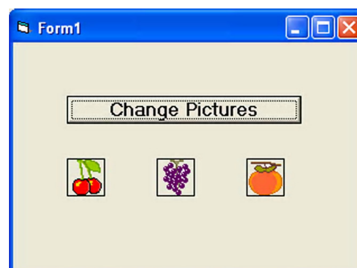


Table 6.4 summarizes the usage patterns of command buttons.

***Table 6.4** Usage Patterns of Command Buttons*

Name of the Buttons	Icon	Action
Standard Command Buttons	Yes	The standard command buttons are used to initiate an immediate action.
Lightweight Command Buttons	Print... Print...	A lightweight command buttons are similar to a standard command buttons except its button frame is shown only when mouse hovers the button.
Menu Buttons	Format ▾ Font... Paragraph... Tabs... Language... Frame... Style... Highlight	A menu button refers to a small set of related commands. This button is used when you need a menu for a small set of related commands.
Split Buttons	Open ▾ Open Open read-only Open as copy Open in browser Open with transform Open and repair Show previous versions	The split buttons are used to consolidate a set of variations of a command especially when one of the commands is used most of the time frequently.
Progressive Disclosure Buttons	More >> << Less	The progressive disclosure button is used to show or hide for infrequently used options. Hiding infrequently used options until they are needed is known as progressive disclosure. Double chevrons (either >> or <<) are used to indicate progressive disclosure and they point in the direction in which the revealing or hiding will take place.

Program

Following code is used to display the usage of graphical CommandButton. The 'Go' CommandButton is used to control the animation of selected picture.

```
Dim i As Integer
Dim j As Integer
Private Sub cmdDirection_Click()
  If cmdDirection.Caption ="go back" Then
    cmdDirection.Caption ="go"
     j = -60
  Else
    cmdDirection.Caption ="go back"
     j = 60
  End If
End Sub
Private Sub Form_Load()
  Set pic.Picture = LoadPicture("Tiger.bmp")
  i = 1
  j = 60
End Sub
Private Sub tmr_Timer()
  If i < 8 Then i = i + 1 Else i = 1
  If (pic.Left < Me.Width And j > 0) Or (pic.Left > 0 And
  j < 0) Then
    pic.Left = pic.Left + j
    Set pic.Picture = LoadPicture("Tiger" & i & ".bmp")
  End If
End Sub
```

The output is as follows:



NOTES

Some of the controls, such as cmdDirection (CommandButton, Caption = "go back"), tmr_Timer (Timer, Interval = 20), pic (PictureBox, AutoSize = -1 'True', BorderStyle = 0 'None') are set in the Property dialog box. To create the graphics image on CommandButton, following program is required:

```
Public Sub InitializeComponent()
      btnHello = New Button()
```

```
btnHello.Text="Hello"
btnHello.Location = New Point(32, 20)
btnHello.Size = New System.Drawing.Size(120, 48)
btnHello.TextAlign = ContentAlignment.BottomCenter
btnHello.Image = Image.FromFile("E:\Programs\Hello.gif")
btnHello.ImageAlign = ContentAlignment.TopCenter
Controls.Add(btnHello)
End Sub
```

The output is as follows:



Instead of using the `Image` property, add a picture from an image list. Then, the `ImageList` property is assigned to the `CommandButton`. The `ImageIndex` property is used to specify the selected picture which will be displayed on the `CommandButton`.

6.6 SUMMARY

- Technically, the Web graphics is the methodology used to create dynamic graphics for the Website. Creating Web graphics is very easy. You can create desired Web graphics or images in Visual Basic or Adobe Photoshop and can upload it to a Web server on the Internet.
- The terms image and Web graphic are often used interchangeably. Once a Web graphic or image is created and uploaded to a Web server on the Internet, it can be linked using HTML or CSS.
- A Web page is an electronic document uniquely written in HTML (HyperText Markup Language). Web pages can include text, graphics, video, animation, sound and interactive elements, such as data entry forms. Web pages also include hyperlinks to other Web pages.
- A Website contains one or more Web pages that share to a common theme, such as a person, business, organization or a subject. The first page of the Website is called the home page, which is an index of the content available on the Website.

NOTES

Check Your Progress

19. Fill in the blanks with appropriate words.
    (a) Web includes Web pages that are basically ________ documents and these pages can be accessed across the network.
    (b) Each pixel on the display surface has a finite size depending on the screen ________ and hence a pixel cannot represent a single mathematical point.
    (c) A Web page may incorporate elements from other Websites with suitable markup ________.
    (d) A computer animation ________ can be set up by specifying the storyboard, the object definitions and the image frames.
20. State whether the following statements are true or false.
    (a) Computer graphics does not include the creation, storage and manipulation of pictures and drawings using a digital computer.
    (b) Static Web pages are fixed content documents which perpetually provide the same information in response to all download requests from all Web users.
    (c) A Website is a collection of related Web pages containing images, videos or other digital files.
    (d) The eyes retain an image long enough to allow the brain to connect the frames in an interrupted sequence.

- Decorative images are also sometimes used to create the image bits that HTML and CSS can not create, for example, diagonal lines and rounded corners. Decorative images are also used for repeating backgrounds on Web pages.
- Computer graphics refers to the creation, storage and manipulation of pictures and drawings using a digital computer.
- In Visual Basic 6.0, the default unit of measurement is known as twips. One twip represents 1/1440 inch or 17.639 µm. Twips represent screen independent units which ensure that the proportion of screen elements is arranged as the same on all the display systems and devices.
- In Visual Basic, the `AutoRedraw` property is used when the graphics methods are called from an event other than the Paint event whereas the `ClipControls` property is used to control the painting of a Form. When the value is set to True, only exposed areas are repainted.
- The WWW is a subset of the Internet and comprises of a huge collection of documents stored in computers across the world.
- The programs that run at the client-side are known as the active documents. Whenever a Web client requests for an active document, the Web server provides a copy of the same in the form of byte code.
- Home page is known as the first page of the Web page. It is replete with a myriad of hyperlinks on its page.
- Launching the Website is an important operation. This requires information pertaining to name, phone, URL description as well as the domain details. Launch of the site is carried out after designing and completion of the site. It is essential to finalize the layout and style of the site before launching. It is significant to note that before the launch of the Website, its domain should be registered.
- A Website is launched with the help of File Transfer Protocol (FTP). It is an economical option. The owner of Website must instruct the Web designer and the system analysts to implement FTP before the launch of the Website.
- A Website is written in HTML and is a collection of linked Web pages on a Web server which can be electronically accessed.
- A static Website is one that has Web pages stored on the server in the format that is sent to a client Web browser. It is primarily coded in HTML.
- A dynamic Website is one that changes or customizes itself frequently and automatically based on certain criteria. Dynamic Websites can have two types of dynamic activities namely, code and content. Dynamic code is invisible or behind the scenes and dynamic content is visible or fully displayed.
- Navigating the Web means moving from one Web page to other. Web page is designed generally in that way in which whole information might come within a page.
- The basic Web architecture is characterized by two-tiered services in which Web client gets the information and Web server provides information to the client.

NOTES

- A Web hosting service refers to a type of Internet hosting service that allows individuals and organizations to make their own Websites accessible via the WWW.

NOTES

6.7 ANSWERS TO 'CHECK YOUR PROGRESS'

1. A major application of computer graphics is in designing engineering and architectural systems.
2. Twips represent screen independent units which ensure that the proportion of screen elements is arranged as the same on all the display systems and devices.
3. A pixel is defined as the smallest size object or color spot that is used to display images on a monitor.
4. The internal surface of the monitor screen is coated with red, green, blue phosphor material that glows when struck by a stream of electrons. This coated material is arranged into an array of millions of tiny cells as red, green and blue, called dots.
5. The aspect ratio of the image is the ratio of the number of X pixels to the number of Y pixels.
6. The `ClipControls` property is used to control the painting of a form. When the value is set to True only exposed areas are repainted.
7. Line drawing is accomplished by calculating intermediate point coordinates along the line path between two given end points.
8. A Web page normally incorporates the basic information and links to navigate in the Websites to which it belongs. Documents in the World Wide Web are classified into three types, namely static, dynamic and active documents.
9. A static Web page is fixed content documents which perpetually provide the same information in response to all download requests from all Web users. Static documents are stored in a Web server to be accessed by the Web client.
10. A Website is a collection of related Web pages containing images, videos or other digital files. It is hosted on at least one Web server, accessible via a network, such as the Internet or a private local area network through an Internet address known as a Uniform Resource Locator or URL.
11. A blog is a type of Website or part of a Website supposed to be updated with new content from time to time
12. The basic Web architecture consists of two-tiers.
13. Animation makes graphics more realistic by imparting motion and dimension to an inanimate object.
14. The `Enable` property determines whether the `CommandButton` is active. You can also change the `Enable` property at runtime with code to prevent the user pressing the button.

15. The simplest way to create animation is to set the `Visible` property of a group of images or pictures or texts and labels to True or False by triggering a set of events, such as clicking a button.
16. `Image.Left` gives the distance of the image in twips from the left border of the screen and `Image.Top` gives the distance of the image in twips from the top border of the screen, where 1 twip is equivalent to 1/1440 inch.
17. The `Font` property produces a Font dialog box in which you can set the caption's font name, style and size.
18. The `ImageIndex` property is used to specify the selected picture which will be displayed on the `CommandButton`.
19. (a) Linked; (b) Resolution; (c) Anchors; (d) Sequence.
20. (a) False; (b) True; (c) True; (d) False.

NOTES

6.8 QUESTIONS AND EXERCISES

Short-Answer Questions

1. What is Web graphic?
2. What is the significance of computer graphics?
3. What is the significance of dot pitch?
4. What does `DrawMode` property controls?
5. Why the `Pen` object is used?
6. Write the first required step to create an animation in Visual Basic.
7. How twips are used to measure distance of the image?
8. Why split buttons are used?
9. What is a Web page?
10. Define Website and its types.

Long-Answer Questions

1. Discuss the significance of Web graphics and its various types with the help of examples.
2. Explain the usage and significance of computer graphics with the help of examples.
3. Describe the various graphics properties and methods with the help of examples.
4. Explain the line, circle, box and ellipse drawing with the help of Visual Basic syntax and coding.
5. Explain the various types of Web pages.
6. Summarize the factors that should be taken into consideration while creating and launching a Website.

NOTES

7. Discuss Web hosting with the help of examples.
8. Explain the significance of animation in Visual Basic graphics.
9. Write a Visual Basic code for interface design which shows the effect of the bird flapping its wing and flying at the same time.
10. Discuss the various usage patterns of `CommandButton`.

Model Question Paper MBA Degree Examination

Internet Programming and Web Design

Time: 3 Hours **Maximum: 100 Marks**

PART A (5 × 8 = 40 marks)

Answer any FIVE of the following:

1. What is the significance of the Internet?
2. Trace the evolution of the Internet and its various technologies.
3. Explain the applications of Perl and CGI variables.
4. Compare and contrast JavaScript and VBScript. Give examples.
5. Why is Internet markup languages used? How it is useful for normal running of Internet, e-mail and RMI?
6. Explain the significance of ActiveX controls.
7. Why Web graphics are used for preparing Websites?
8. Trace the recent developments in the Internet based e-mail technologies. How it is beneficial for Internet users?

PART B (4 × 15 = 60 marks)

Answer any FOUR of the following:

9. Explain the major features and working of the Internet. Demonstrate the inter linkages amongst the various elements of the Internet.
10. Classify the various Internet programming languages giving significant features of each. Explain the scope and importance of each designing a Website or a Web page.
11. Explain the various types of Internet scripting languages and their applications.
12. Why the Internet markup languages are gaining importance? Justify your answer giving examples.
13. Explain the various types of ActiveX controls with the help of examples. List the areas where these controls are used.
14. Discuss the significance of Web design for creating a successful and interactive Website.

Compulsory

15. Write a HTML code to create a client-side image map for the OBJECT element and hide the MAP element within the OBJECT element's content such that the MAP element's contents will only be rendered if the OBJECT cannot be rendered.